



Universidad de las Ciencias Informáticas

Facultad # 8

*“Propuesta de Documentación del Proceso de Desarrollo de
Software Educativo”*

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Lisandra del Carmen Hernández Cuello

Tutora: Ing. Yaillet Martínez Pérez

Ciudad de La Habana

Junio 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

Curso: 2008-2009

"No hay más que un camino para el progreso en la educación, como en todas las cosas humanas, y es el de la ciencia guiada por el amor. Sin ciencia, el amor es impotente; sin amor, la ciencia es destructiva."

Bertrand Russell, 1979.



Quiero agradecerle:

A mami y papi, por el gran amor incondicional que me profesan, por el apoyo que me han brindado durante el transcurso de mi carrera en la UCI. A ustedes les debo todo lo que soy hoy en día. Gracias por confiar en mí.

A Leo (mi hermano), gracias a ti he llegado a ser la Ingeniera en Ciencias Informáticas que tanto soñaron nuestros padres.

A Angelito (mi hermanito), espero que Leo y yo te sirvamos de ejemplo para tu vida futura tanto en lo personal como en lo profesional.

A mi padrastro Rafael, por la preocupación y el apoyo que me ha brindado en todo momento.

A mis amigas, por el apoyo que siempre me han brindado en los momentos más difíciles en la UCI: el Plieque (Yaniris), la Boya (Yanet), Tulina (Adriana), la Elástica (Odalis), la China (Anny), Yaimis, Liniel, Chuchi (Adriana Forns), Caza Manguitos (Anita), Bola de Pelos (Dairy), Lianne, las Danias. A todas mis chicas que de una forma u otra me apoyaron muchísimo para la realización de mi trabajo de diploma... GRACIAS.

A mi amiga Dayana que compartió mis inicios en la UCI y gran parte de mi carrera, así como a mi amiga Lisbeth que ha sabido compartir su amistad conmigo todos estos años incluyendo los del Pre-Universitario, y ha sido más que mi amiga, mi hermana.

A mi tía Ana Rita y a mis primos (Luis y Diana), gracias por el apoyo que me han brindado en todo momento.

A mi tío Jorgito, mi primita Michelle, que a pesar de la lejanía siempre estuvieron al tanto de mí y de Leo preocupándose por nuestros estudios.



A mi tutora por el apoyo brindado durante todo el desarrollo de este trabajo investigativo.

A los profesores que me han ayudado incondicionalmente en los momentos que más los necesité: Liana Isabel Araujo Pérez y Lannie Octavio Herrera Pérez.

A todos los que al verme a diario me preguntaban: “¿Cómo va la tesis?”.

A todos un millón de gracias, espero tener la dicha de servirles siempre.

Lisy



Quiero dedicarle el presente Trabajo de Diploma:

Ante todo a mi abuela, que fue y seguirá siendo la inspiración de mi vida y la luz en mi camino.

Siempre serás mi Ángel Guardián. Mi Chula TE AMO.

A mi madre y mi padre que son lo más grande que tengo en esta vida. Gracias por servirme de ejemplo, gracias por existir y estar siempre a mi lado. Los Amo.

A mi hermano Leandro, que sin su perseverancia, sus regaños, su empeño, no hubiese logrado este sueño de convertirme en lo que soy hoy, una INGENIERA.

A los que se quedaron en el camino, pero tuvieron la fuerza, valentía y el coraje de intentarlo.



Declaro que soy la única autora del presente trabajo el cual lleva por título Propuesta de Documentación del Proceso de Desarrollo de Software Educativo; y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autora: Lisandra del Carmen Hernández Cuello

Tutora: Ing. Yaillet Martínez Pérez



Resumen

El Software Educativo (SWE), es una de las principales líneas de desarrollo en Cuba y especialmente en la Universidad de las Ciencias Informáticas (UCI). Precisamente la documentación del proceso de desarrollo de software (SW) no se ajusta a los proyectos de SWE provocando demoras en su organización y entrega, influyendo en la eficiencia del mismo. En el presente trabajo investigativo se propone un conjunto de documentos para el proceso de desarrollo del SWE sirviendo de apoyo y ayuda para mantener de manera organizada la información almacenada en cada una de sus fases. Para ello se realizó un estudio de las diferentes metodologías de desarrollo de software justificando cada uno de los documentos seleccionados para establecer un adecuado procedimiento para el desarrollo del mismo. Con la finalidad de validar la propuesta planteada en la investigación se efectuó un análisis detallado del Método de Expertos conformado por especialistas en materia de SWE, quienes tendrán la tarea de aceptar o rechazar lo expuesto en el trabajo.

Palabras Clave: Software Educativo, Metodologías de Desarrollo.



Índice

Introducción	1
Capítulo#1: Fundamento Teórico	6
1.1 INTRODUCCIÓN.....	6
1.2 SOFTWARE EDUCATIVO	6
1.2.1 Desarrollo del Software Educativo en Cuba	9
1.3 MODELOS DE DESARROLLO DE SOFTWARE EDUCATIVO.....	10
1.4 ESTUDIO DE LOS PROCESOS O METODOLOGÍAS EXISTENTES	11
1.5 ALGUNOS ENFOQUES DE DOCUMENTACIÓN PARA LOS PROCESOS DE DESARROLLO DE SOFTWARE	13
1.6 CONCLUSIONES DEL CAPÍTULO.....	20
Capítulo#2: Desarrollo de la Propuesta	21
2.1 INTRODUCCIÓN.....	21
2.2 MODELACIÓN CONCEPTUAL.....	21
2.2.1 Documentación Necesaria.....	22
2.3 GESTIÓN DE REQUISITOS.....	23
2.3.1 Documentación Necesaria.....	24
2.4 ANÁLISIS Y DISEÑO DEL SISTEMA.....	26
2.4.1 Documentación necesaria para ambos procesos.....	27
2.5 DISEÑO GRÁFICO	28
2.5.1 Documentación Necesaria.....	28
2.6 GESTIÓN AUDIOVISUAL.....	28
2.6.1 Documentación Necesaria.....	29
2.7 IMPLEMENTACIÓN.....	29
2.7.1 Documentación Necesaria.....	30
2.8 PRUEBA	31
2.8.1 Documentación Necesaria.....	32
2.9 IMPLANTACIÓN	33
2.9.1 Documentación Necesaria.....	34



2.10 CONCLUSIONES DEL CAPÍTULO	36
Capítulo#3: Validación de la Propuesta.....	37
3.1 INTRODUCCIÓN.....	37
3.2. MÉTODO DE EXPERTO	37
3.2.1 Aplicación del Método Experto	37
3.3. CONCLUSIONES DEL CAPÍTULO	43
Conclusiones Generales.....	44
Recomendaciones	45
Referencias Bibliográficas	46
Anexos	49
ANEXO #1: DOCUMENTOS DE SALIDA DE LOS PROCESOS DEL CICLO DE VIDA	49
ANEXO #2: ETAPAS DEL CICLO DE VIDA	52
ANEXO #3: PLANILLA MODELO PEDAGÓGICO	55
ANEXO #4: PLANILLA DOCUMENTO VISIÓN	59
ANEXO #5: PLANILLA ESPECIFICACIÓN DE REQUISITOS	67
ANEXO #6: PLANILLA GLOSARIO DE TÉRMINOS.....	73
ANEXO #7: PLANILLA DESCRIPCIÓN PARA LA ARQUITECTURA DE SOFTWARE.....	75
ANEXO #8: PLANILLA MAPA DE NAVEGACIÓN	80
ANEXO #9: PLANILLA GUIÓN DE CONTENIDOS	83
ANEXO #10: PLANILLA DISEÑO DE PRUEBAS PEDAGÓGICAS	87
ANEXO #11: PLANILLA ESPECIFICACIÓN DE CASOS DE USO.....	93
ANEXO #12: PLANILLA PLAN DE INTEGRACIÓN.....	96
ANEXO #13: PLANILLA PLAN DE ITERACIÓN	98
ANEXO #14: PLANILLA PLAN DE PRUEBAS.....	100
ANEXO #15: PLANILLA MANUAL DE INSTALACIÓN.....	109
ANEXO #16: PLANILLA MANUAL DE USUARIO.....	113
ANEXO #17: PLANILLA PLAN DE ADIESTRAMIENTO.....	115
ANEXO #18: PLANILLA PLAN DE IMPLANTACIÓN	120



ANEXO #19: PLANILLA PAUTAS DEL DISEÑO GRÁFICO	125
ANEXO #20: PLANILLA PLAN AUDIOVISUAL	129
ANEXO #21: MODELO PARA INFORMAR EL PESO DE LOS CRITERIOS.....	133
ANEXO #22: MODELO PARA CALIFICAR LOS CRITERIOS.	134



Índice de Tablas

Tabla 1: Características principales de los programas educativos. (Marqués, 1998).....	6
Tabla 2: Modelación Conceptual	14
Tabla 3: Gestión de Requisitos.....	14
Tabla 4: Análisis y Diseño del Sistema.....	15
Tabla 5: Implementación.....	16
Tabla 6: Prueba	17
Tabla 7: Implantación.....	18
Tabla 8: Peso Promedio/Criterio.....	39
Tabla 9: Consistencia en Trabajo de Expertos.	40
Tabla 10: Cálculo de P*c.	41
Tabla 11: Documentos de salida de los procesos del ciclo de vida. (Cataldi, 2000)	49
Tabla 12: Etapas del ciclo de vida	52



Introducción

En la actualidad, con el gran desarrollo científico – técnico se necesita de hombres que sean capaces de consultar una amplia gama de información en poco tiempo; y que tengan además conocimiento suficiente para solucionar problemas de forma innovadora. Resulta innegable el incremento de las tecnologías de la informática y las comunicaciones en las diferentes esferas de la sociedad a escala mundial. El gran desarrollo de las ciencias y la tecnología ha llevado a la sociedad a entrar en un nuevo milenio llamado: “Era de la informatización”, sin dudas se está en presencia de una revolución tecnológica y cultural de alcances insospechados.

“En el tránsito del siglo XX al XXI Cuba ha vivido momentos importantes de su desarrollo educacional.” El país realiza significativas transformaciones en todos los niveles del sistema nacional de educación, en un proceso de cambio que tiene como marco político e ideológico la llamada *Batalla de Ideas*, cuyo objetivo final es elevar, de manera continua la calidad de la educación. Se trata de procesos de innovación que abarcan principios básicos de la educación, formas de trabajo pedagógico escolar, variantes de interpretación del currículo y estrategias para la educación ciudadana y el aprendizaje. (Ruz, 2004)

Durante los últimos años, Cuba ha evolucionado en la esfera de desarrollo del software. Fundamentalmente, las empresas han comenzado la automatización e informatización de diferentes áreas, donde se desempeña, en un corto período de tiempo y con una mejor calidad, un mejor uso de las tecnologías. Debido a esto, la Industria Cubana del Software, se ha trazado una serie de tareas primordiales para informatizar la sociedad y adentrarnos en el Mercado de software a nivel universal.

La informática está llamada a convertirse en un campo boyante internacionalmente, en una de las ramas más productivas para el país (Ruz, 2004). Para dominar esta meta es preciso lograr el respaldo de un consistente sistema de educación superior donde es significativo el lugar que ocupa la Universidad de las Ciencias Informáticas (UCI).

La UCI, creada hace 7 años, iniciando el curso escolar el 23 de Septiembre de 2002, es una de las instituciones del país donde se forman cientos de jóvenes como especialistas informáticos del nivel superior con un sistema avanzado de educación, investigación y producción, quienes desarrollan disímiles



tipos de software, entre ellos los software educativo, siendo un perfil principal de producción de la universidad.

Sobre el objetivo de este centro Fidel Castro Ruz planteó: “El propósito fundamental es lograr un centro de excelencia para la formación masiva de profesionales de nivel superior. Ello debe alcanzarse con la ejecución de ambiciosos programas curriculares y de producción y con la aplicación de las más modernas tecnologías en la docencia.” (Ruz, 2004)

En la universidad ya se han dado los primeros pasos para la introducción paulatina de estas buenas prácticas para el desarrollo de software. En la investigación se abordará específicamente el proceso de desarrollo para SWE. Software que va a presentar como objetivo principal, lograr que el aprendizaje se convierta en un proceso natural y permanente para los estudiantes y docentes. Para ello se hace necesario aprender a usar las Nuevas Tecnologías de la Informática y usar a su vez dichas tecnologías para aprender.

Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida. Según Piattini considera a la metodología como *“un conjunto de pasos y procedimientos que deben seguirse para el desarrollo del software”*. (Piattini, 1996)

La aplicación de metodologías es primordial para la confección de productos de calidad, pues estos precisan un conjunto de criterios que rigen al proceso de desarrollo de software. Muchas son las instituciones que se rigen por modelos de calidad, implantando exigencias o necesidades que deben cumplir para lograr un buen desarrollo y servicio del software, y así de este modo mantener y alcanzar un producto de buena calidad.

Como ejemplo de esos modelos se encuentra el Modelo Integrado de Capacidad y Madurez (CMMI), quien en su versión 1.2 tiene como objetivo específico “Establecer activos de proceso de la organización” dominando un conjunto de prácticas y subprácticas específicas, centrándose la investigación en “Documentar los procesos de la organización”. Estas subprácticas permitirán tener un control y seguimiento de los pasos a realizar durante todo el proceso de desarrollo del software a través de cada uno de los niveles de madurez por los que atraviesa la organización, por ende será una documentación densa y voluminosa por el trabajo de mantenimiento que ello implica.



El Software Process Improvement and Capability Determination (SPICE) es otro de los modelos de calidad fundamentado en un conjunto de prácticas. El desarrollo de la presente investigación se centrará en “Documentar el proceso estándar”, la cual permitirá documentar los estándares, procedimientos y modelos para realizar el proceso y para caracterizar sus salidas.

En estudios realizados a proyectos productivos fundados en la UCI se arribó a la conclusión de que no existen expedientes de proyectos que se ajusten específicamente al proceso de desarrollo para SWE. Dichos expedientes carecen de documentos o artefactos que pauten toda la información de la parte pedagógica del software; es decir, pueden ser aplicados a cualquier software que se desarrolle en la universidad. Todo esto puede traer consigo que no se mantenga una correcta organización y planificación durante el desarrollo del proyecto causando una insuficiente calidad a través de todo el proceso educativo. Para ello la investigación se apoya en las metodologías de desarrollo de software existentes con el objetivo de fundamentar y adaptar los documentos necesarios para darle seguimiento a los pasos que deben seguirse para el desarrollo del SWE.

Problema a resolver:

La documentación definida para los proyectos de software en la UCI no es adaptable para documentar los proyectos de software educativo.

Objeto de Estudio:

El proceso de desarrollo de los proyectos de software educativo.

Campo de Acción:

Documentación que se ha de generar en los proyectos de software educativo.

Idea a defender:

Una propuesta para la documentación de los proyectos de desarrollo de software educativo ayudará a la eficiencia del proceso de desarrollo.



Objetivo General:

Proponer una documentación para el proceso de desarrollo del Software Educativo que permita mantener una correcta organización de la información almacenada en las fases de desarrollo del software.

Objetivos Específicos:

- Desarrollar un estudio detallado de bibliografías relacionadas al tema de la investigación.
- Realizar una propuesta de documentación para proyectos de software educativo en la UCI.
- Aprobar la propuesta realizada a través del estudio del Método de Evaluación de Experto.

Para el cumplimiento de los objetivos específicos antes mencionados, se propone la realización de las siguientes tareas:

Tareas Investigativas:

- Identificar los procesos o actividades por los que se guiará la propuesta.
- Evaluar la documentación existente para cada uno de los procesos del desarrollo de software educativo.
- Desarrollar una propuesta de documentación para cada fase del desarrollo de software educativo.
- Evaluar la propuesta por medio del Método de Evaluación de Experto.

Métodos Científicos:

Estos métodos de investigación se han utilizados para la obtención, análisis y llegada a conclusiones. Dentro de los cuales podemos ver los siguientes:

- **Métodos Teóricos:**
 - Analítico-Sintético: Permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio. Establece mentalmente la unión entre las partes



previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas.

- Histórico-Lógico: Utilizado para el estudio profundo de los trabajos anteriores y tomando éstos como punto de partida; comparándolos y analizar profundamente los resultados obtenidos.
- Genético: Consiste en determinar el campo de acción elemental que se convierte en célula del objeto. En dicha célula están representados todos los componentes del objeto, así como sus leyes más trascendentales.

- **Métodos Empíricos:**

- Observación: Se realiza un análisis profundo a las metodologías de desarrollo para orientar el objetivo del trabajo presentado.
- Encuestas: Se realiza un cuestionario preelaborado a un grupo de personas para adquirir información relacionada con el proceso de desarrollo de SWE.



Capítulo#1: Fundamento Teórico

1.1 Introducción

En el presente capítulo se fundamentará el tema de la investigación, abordando conceptos y definiciones vinculados al software educativo. Se presentará un estudio detallado de las distintas metodologías existentes para el proceso de desarrollo de software seleccionando de estas los documentos necesarios para el proceso educativo. De esta manera quedará plasmada la propuesta de la investigación.

1.2 Software Educativo

Debido a los adelantos científicos y tecnológicos, el cambio se ha hecho evidente en todos los campos del saber. De manera especial en las comunicaciones, la biomedicina y desde luego las ingenierías de sistemas y electrónica. Esto ha afectado significativamente las tradicionales formas de enseñanza-aprendizaje, que se han venido utilizando desde los últimos 30 años en que apareció el software educativo.

Marqués (1995) sostiene que se pueden usar como sinónimos de "software educativo" los términos "programas didácticos" y "programas educativos", centrando su definición en "aquellos programas que fueron creados con fines didácticos, en la cual excluye todo software del ámbito empresarial que se pueda aplicar a la educación aunque tengan una finalidad didáctica, pero que no fueron realizados específicamente para ello".

Tabla 1: Características principales de los programas educativos. (Marqués, 1998)

Características	Descripción
Facilidad de uso	En lo posible autoexplicativos y con sistemas de ayuda
Capacidad de motivación	Mantiene el interés de los alumnos
Relevancia curricular	Relacionados con las necesidades del docente
Versatilidad	Adaptables al recurso informático disponible
Enfoque pedagógico	Que sea actual: constructivista o cognitivista.
Orientación hacia los alumnos	Con control del contenido del aprendizaje



Evaluación	Incluirán módulos de evaluación y seguimiento.
------------	--

En la Tabla 1 se pueden observar algunas de las características principales de los programas educativos. Se da por sentado que los programas deben usarse como recursos que incentiven los procesos de enseñanza y de aprendizaje, con características particulares respecto de otros materiales didácticos y con un uso intensivo de los recursos informáticos de que se dispone. (Marqués, 1998)

La clasificación de los Software Educativos depende de diversos elementos, entre los cuales está la forma de diseñar, presentar y utilizar las actividades de aprendizaje. Además es relevante para qué actividad pedagógica se utilizará el software. Entre los distintos tipos de software educativo están:

- Ejercitación
- Tutorial
- Simulación
- Juego educativo
- Material de referencia multimedia
- Historias y cuentos
- Editores
- Hiperhistoria

El software educativo se caracteriza por ser muy interactivo mediante el empleo de recursos multimedia, ejemplo: videos, sonidos, fotografías, diccionarios científicos, fundamentos de profesores experimentados, ejercicios y juegos interactivos e instructivos que apoyan el sistema de evaluación. Además pueden tratar las diferentes materias (Matemática, Idiomas, Geografía, Dibujo), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y con más o menos posibilidades de interacción; pero todos comparten las siguientes características:



- Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Facilita las representaciones animadas.
- Incide en el desarrollo de las habilidades a través de la ejercitación.
- Permite simular procesos complejos.
- Reduce el tiempo de que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- Facilita el trabajo independiente y a la vez un tratamiento individual de las diferencias.
- Permite al usuario (estudiante) introducirse en las técnicas más avanzadas.

Las funciones del software educativo, están determinadas de acuerdo a la forma de uso de cada profesor. A continuación, se describen algunas de las funciones que pueden realizar los programas según (Marqués, 1995):

- ✓ **Informativa:** Presentan unos contenidos que proporcionan una información estructuradora de la realidad. Representan la realidad y la ordenan. Son ejemplos, las bases de datos, los simuladores, los tutoriales.
- ✓ **Instructiva:** Promueven actuaciones de los estudiantes encaminadas a facilitar el logro de los objetivos educativos, el ejemplo son los programas tutoriales.
- ✓ **Motivadora:** Suelen incluir elementos para captar el interés de los alumnos y enfocarlos hacia los aspectos más importantes de las actividades.
- ✓ **Evaluadora:** Al evaluar implícita o explícitamente, el trabajo de los alumnos
- ✓ **Investigadora:** Los más comunes son: las bases de datos, los simuladores y los entornos de programación.
- ✓ **Expresiva:** Ya que el entorno informático, no permite ambigüedad expresiva.



- ✓ **Metalingüística:** Al aprender lenguajes propios de la informática.
- ✓ **Lúdica:** A veces, algunos programas refuerzan su uso, mediante la inclusión de elementos lúdicos.
- ✓ **Innovadora:** Cuando utilizan la tecnología más reciente.

Todas estas funciones antes explicadas son las que de una forma u otra van dando un marco pedagógico-funcional al software educativo, consiguiendo que sea de fácil uso y entendimiento para el usuario. Estas funcionalidades serán las encargadas de darles una vía más factible a los profesores para lograr una apropiada relación estudiante-profesor.

1.2.1 Desarrollo del Software Educativo en Cuba

Antes del surgimiento del SWE en Cuba era imposible pensar en el desarrollo de la educación; sin embargo, actualmente los avances tecnológicos permiten el empleo del imprescindible medio informático en las diferentes enseñanzas.

En Cuba se cuenta con tres colecciones de gran importancia para maestros y alumnos: *Multisaber*, — primer gran esfuerzo nacional dedicado a la Enseñanza Primaria—; *El Navegante*, para la Secundaria Básica; y *Futuro*, diseñado para el Bachillerato, la Enseñanza Técnica-Profesional y la Educación de Adultos. (Valenzuela, 2008)

La introducción de este medio informático dentro del proceso de enseñanza aprendizaje tiene su expresión en el preuniversitario en la colección Futuro; de la cual es representante el software Nuestro Planeta, como complemento informático del desarrollo del programa de la asignatura Geografía General para el décimo grado. A diferencia de otros productos, la concepción pedagógica de este software basada en el hiper-entorno de aprendizaje unido a una navegación amigable, hacen de este una aplicación informática muy completa. Dicha aplicación sirve para el uso de estudiantes, docentes y de cualquier persona interesada en incrementar su cultura geográfica y medio ambiental. (Valenzuela, 2008)

Numerosos son los logros que se han alcanzado hasta el momento como resultado de los grandes esfuerzos realizados por el gobierno cubano, al punto que se puede decir que las Tecnologías de la Información y de las Comunicaciones (TIC) se han insertado en casi todas las ramas de nuestra sociedad.



Actualmente se sigue perfeccionando el trabajo y ampliando el radio de acción de las nuevas tecnologías en beneficio de todas las personas.

Ejemplo de ello es la Industria Cubana del Software, que con la participación de la UCI y otras empresas productoras de software del país está dando grandes pasos. Este centro de estudios universitarios es un proyecto nacido de la Revolución Cubana, denominado inicialmente "Proyecto Futuro", con dos objetivos: informatizar el país y desarrollar la industria del software para contribuir al desarrollo económico del mismo. Entre los principales productos que desarrolla tenemos el software educativo con el objetivo de mejorar la calidad de la formación integral del estudiante en todo el país. Para ello se hace necesario ajustar la documentación existente por otros proyectos productivos al SWE con la finalidad de alcanzar una buena organización de la información y lograr un eficiente proceso de desarrollo.

1.3 Modelos de Desarrollo de Software Educativo

Cuando se va a desarrollar un software se puede emplear cualquiera de los diferentes modelos de desarrollo, entre los cuales tenemos:

- El Modelo Integrado de Capacidad y Madurez (CMMI), modelo de referencia de prácticas maduras usadas para evaluar y mejorar la capacidad de los procesos.
- Modelo Lineal (Modelo en Cascada) el cual presenta una estructura secuencial (de ahí el nombre de Modelo en cascada) formada por seis fases o etapas, que se desarrollarán de forma secuencial.
- Modelo de Construcción de Prototipos, se desarrolla de manera no secuencial. Está basado en la construcción de simulaciones o modelos ejecutables de aplicaciones más extensas, y su objetivo principal está enmarcado en la participación directa del cliente en la construcción del software requerido.
- Modelo en Espiral: Es un modelo para superar algunas de las limitaciones del modelo en cascada. La espiral se forma a partir de una serie de ciclos de desarrollo y va evolucionando. Los ciclos internos de la espiral denotan análisis y prototipado y los externos el modelo clásico. En la dimensión radial están los costos acumulativos y la dimensión angular representa el progreso realizado en cada etapa. (Boehm, 1988)



Específicamente se abundará más acerca de los modelos de desarrollo de SWE, quienes implican el planteamiento de los diferentes aspectos que componen este proceso, de las personas que participan en él y de los procedimientos y etapas que lo componen. Cada uno de estos elementos se presenta, además, con las relaciones que se pueden establecer en ellos y las diferentes alternativas que se pueden utilizar para su representación.

1.4 Estudio de los procesos o metodologías existentes

Son múltiples las propuestas de modelos de desarrollo de software que se encuentran para la producción de software educativo, la mayoría de ellas dirigidas al desarrollo de software prototípico, elaborado por grupos especializados de profesionales.

Como ejemplo de estos modelos están Dick y Carey (1978) y (Gagné, 1987), quienes propusieron un *modelo sistemático* de tipo *estructurado* basado en teorías instruccionales; mientras que Tripp y Bichelmayer (1990) trabajaban en base a las de tipo *no lineal*. Luego aparecen los modelos orientados al desarrollo de software multimedial, propuesto por Álvaro Gálvis (1992), basado en la ingeniería de software, Vélez et al. (1997), basado en procesos técnicos de cinco fases, Zambrano (1995), inicia la etapa de desarrollo gráfico y programación y el estudio del producto, o evaluación de calidad; y muchos otros que desarrollaron metodologías para el software educativo hipermedial.

Para la fundamentación de la propuesta fueron analizadas algunas metodologías para el desarrollo de software, como por ejemplo Rational Unified Process (RUP); y específicamente para el SWE la metodología de Pere Marqués y Zulma Cataldi.

RUP propone que para desarrollar un software se necesita una forma coordinada de trabajo, un proceso que integre las múltiples facetas del desarrollo, cuyo objetivo será producir software de alta calidad; es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido, cubriendo el ciclo de vida y desarrollo de software. Con el objetivo de lograr un mejor seguimiento y supervisión durante el desarrollo del software, surge el *Expediente de Proyecto*, documento que tendrá como la finalidad recopilar todo el proceso del proyecto desde su concepción hasta su finalización, poniendo especial énfasis en cuantificar la obtención de los resultados, tanto a nivel cuantitativo como cualitativo.



Cataldi plantea que la metodología considera la construcción de software educativo desde un aspecto integral, teniendo en cuenta los aspectos pedagógicos en el ciclo de vida. La propuesta metodológica de Cataldi incorpora procesos nuevos que previenen las necesidades pedagógicas – didácticas y define nuevas actividades que cautelen cuestiones pedagógicas – didácticas en los procesos existentes. Marqués es otro de los autores que plantea un ciclo de desarrollo para software educativo de programas en diez etapas, con una descripción detallada de las actividades y recursos necesarios para cada una de ellas.

Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida; es decir no son más que *“un conjunto de pasos y procedimientos que deben seguirse para el desarrollo del software”*. (Piattini, 1996)

Según Maddison en el año 1983 sostiene que dichas metodologías están compuestas por un conjunto de componentes que se especifican a continuación: (Maddison, 1983)

- Cómo se debe dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué salidas se producen y cuándo se deben producir.
- Qué restricciones se aplican.
- Qué herramientas se van a utilizar.
- Cómo se gestiona y controla un proyecto.

Muchos de estos modelos pueden ser utilizados por cualquier tipo de software, solamente se manejarán algunos aspectos para desarrollar SWE, puesto que existen algunos criterios que no son adaptables al mismo.

En el trabajo de maestría presentado por la ingeniera Yaillet Martínez Pérez, el cual lleva por título “Propuesta de procesos para el desarrollo de proyectos de Software Educativo en la Universidad de las Ciencias Informáticas”, se definen cada una de las fases por las que atravesará el proceso de desarrollo del SWE.

Para generar las documentaciones referente al proceso de desarrollo del software educativo, la



investigación se apoya en cada uno de las fases propuestas en el trabajo antes mencionado, con el objetivo de documentar toda la información que fluye durante todo el proceso productivo.

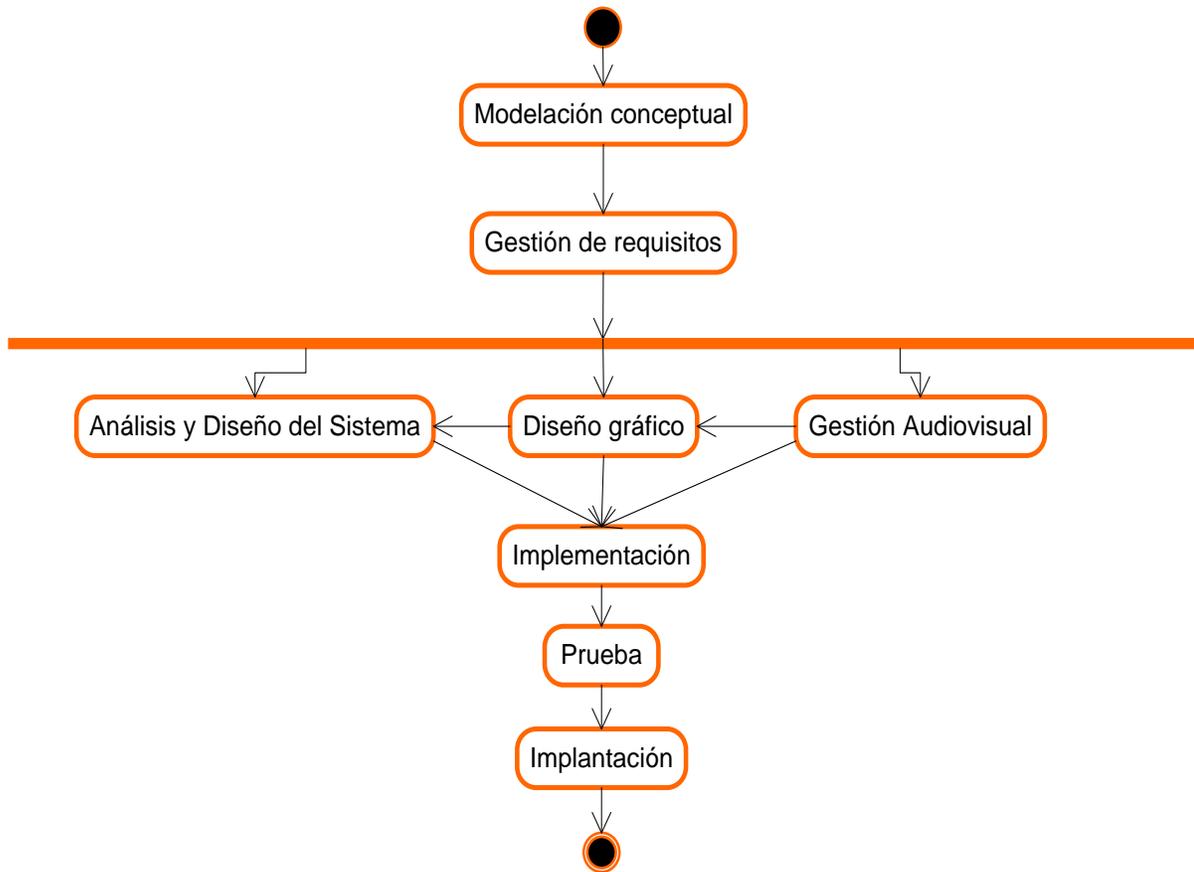


Figura 1. Diagrama de subprocesos que describen el ciclo de desarrollo para el software educativo. (Pérez Y, 2007)

1.5 Algunos enfoques de documentación para los procesos de desarrollo de software

Muchas son las metodologías y numerosos son los ingenieros que abordan el tema de las documentaciones para los procesos de desarrollo de software. A continuación, mostraremos ejemplos de los expedientes que son generados por las propuestas que ofrecen los especialistas en el tema de la investigación, así como los documentos de salida que se crean durante el ciclo de vida de las



metodologías presentadas. En el caso de RUP solamente se tendrán en cuenta los documentos que se generan en los flujos de ingeniería.

Tabla 2: Modelación Conceptual

Metodologías	Documentaciones	Observaciones
RUP	Glosario de Términos	Define términos importantes que se utilizan en el proyecto.
Pere Marqués	No posee	
Zulma Cataldi	Plan de Gestión Plan de Retiro	Enuncian términos concretos para la puesta en marcha del proyecto.
	Informe de Necesidades	Formula posible soluciones y estudia su viabilidad.

La metodología de Pere Marqués no propone documentación alguna referente a la **Modelación Conceptual**; sin embargo RUP formula algunos documentos que no son específicos del SWE, pero sí ofrecen algunos aspectos que son necesarios para la fundamentación de dicho proceso; y Cataldi establece informes que plasman los contenidos necesarios para el arranque del proyecto.

Tabla 3: Gestión de Requisitos

Metodologías	Documentaciones	Observaciones
RUP	Especificación de Requisitos	Especifica detalladamente información acerca de los mecanismos de control que deben utilizarse para controlar los cambios del producto.
	Documento Visión	Define la vista de los interesados del producto que se va a desarrollar.
Pere Marqués	No posee	



Zulma Cataldi	Especificación de requisitos funcionales.	Detalla los requisitos del hardware y software.
	Especificación de interfaces del sistema o programa.	Especifica las interfaces del hardware y con el sistema físico.

En la fase de **Gestión de Requisitos** el ingeniero Pere Marqués no establece una documentación concreta que defina la especificación de los requisitos tanto funcionales como no funcionales del producto. Sin embargo el resto de las metodologías estudiadas establecen documentos que son fundamentales para una correcta selección de los requisitos del sistema.

Tabla 4: Análisis y Diseño del Sistema

Metodologías	Documentaciones	Observaciones
RUP	Especificación de Casos de Uso	Describe cómo se lleva a cabo y se ejecuta un caso de uso determinado.
	Descripción para la Arquitectura del Software	Especifica las ideas principales del diseño y proporciona una descripción entendible de la arquitectura del sistema software.
Pere Marqués	Diseño Funcional	Incluye un esquema del formato y del contenido de la documentación que acompañará al programa. Formado por los siguientes apartados: <ul style="list-style-type: none"> - Ficha Resumen - Manual de Usuario - Guía Didáctica
Zulma Cataldi	<ul style="list-style-type: none"> - Especificación de los objetivos y estructuración de conceptos. - Selección de contenidos y 	Realiza un análisis de los requisitos educativos.



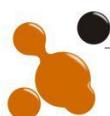
	pertenencia.		
	Diseño Instruccional	Del contenido	<p>Detalla los componentes que conforman el diseño del contenido educativo, identificando los procesos mentales a estimular.</p> <p>Define las actividades a realizar por los alumnos.</p> <p>Jerarquización de los conceptos.</p>
		Del software	<p>Detalla los componentes que conforman el diseño del software y de la arquitectura.</p> <p>Describe el flujo de información, bases, interfaces y algoritmos.</p>

Todas las metodologías abordadas en la investigación, durante esta fase de desarrollo, presentan documentaciones encargadas de almacenar la información referente al proceso de **Análisis y Diseño del Sistema**; a pesar de que algunas serán identificadas por otros procesos. Ejemplo de ello es el documento Diseño Funcional propuesto por Pere Marqués.

En la fase de desarrollo **Diseño Gráfico** y en la **Gestión Audiovisual**, ninguna de las metodologías mencionadas anteriormente define documentos referentes a dicho proceso. Por lo tanto, más adelante en el capítulo 2, se realizará una propuesta de documentación para satisfacer las necesidades de esta fase del SWE.

Tabla 5: Implementación

Metodologías	Documentaciones	Observaciones
RUP	Plan de Integración	Define el orden en que los componentes deben implementarse.
	Plan de Iteración	Define para cada una de las iteraciones a realizarse un conjunto



		de tareas, actividades y recursos.
Pere Marqués	No posee	
Zulma Cataldi	Plan de Integración	Define los procesos de implementación e integra los módulos que deben implementarse.
	Datos para las pruebas	
	Documentación para el sistema o programa y del usuario.	

Para el proceso de **Implementación**, las documentaciones que se generan para esta fase de desarrollo, se apoyan en la metodología RUP ya que es adaptable a cualquier tipo de software. En el caso de Pere Marqués, no se realiza un análisis detallado referente a la implementación del SWE; mientras que Cataldi hace referencia prácticamente a los mismos documentos de RUP.

Tabla 6: Prueba

Metodologías	Documentaciones	Observaciones
RUP	Plan de prueba	Define los objetivos de las pruebas.
	Listado de ideas de prueba	Enumera ideas que identifican las pruebas potencialmente útiles de realizar.
	Listado de datos de prueba (Modelo de datos)	Describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación.
Pere Marqués	No posee	
Zulma Cataldi	Plan de verificación y validación.	Permiten comprobar si el sistema satisface los requisitos del usuario durante el desarrollo del software.
	Plan de pruebas.	
	Especificación y resumen de la prueba.	Ofrece un resumen en cada una de las pruebas realizadas y finalmente afirma si se aprueba o no el producto final.
	Resultado de las Pruebas	



En esta fase de **Prueba**, Marqués no hace alusión a las pruebas que se deben de realizar durante el desarrollo del SWE, a diferencia de Cataldi y RUP que ambos establecen el Plan de Pruebas con la finalidad de comparar los objetivos trazados con los obtenidos. Además se hace necesario un documento que pauté el diseño de pruebas pedagógicas para determinar las debilidades, fortalezas y aplicabilidad en las distintas áreas de enseñanza-aprendizaje del software.

Tabla 7: Implantación

Metodologías	Documentaciones	Observaciones
RUP	Manual de Instalación	Refleja los lineamientos que hay que seguir para instalar el sistema.
	Manual de Usuario	Proporciona una ayuda a las personas que manipularán directamente el producto, acerca del uso que le debe dar al sistema.
	Plan de Adiestramiento	Indica el programa de entrenamiento que será impartido a los usuarios finales para enseñarles el uso, operación y mantenimiento del sistema.
	Plan de Implantación	Define un conjunto de tareas necesarias para poder poner en funcionamiento el sistema en las instalaciones de los usuarios.
Pere Marqués	No posee	
Zulma Cataldi	Plan de instalación.	Se encarga de mostrar los pasos a seguir para la instalación del sistema.
	Informe de instalación.	Recoge cada una de las acciones que se realicen en el proceso de instalación.
	Recomendaciones de mantenimiento.	Muestra los aspectos necesarios para lograr un correcto mantenimiento del sistema.
	Plan de formación y captación	Detalla los aspectos que debe cumplir el usuario final para un uso correcto del sistema.



	Plan de gestión de la configuración	Define un grupo de actividades precisas para la configuración del sistema.
--	-------------------------------------	--

Finalmente durante el proceso de Implantación, la metodología de Marqués no establece una documentación para definir los pasos a seguir para un correcto establecimiento y organización del sistema; sin embargo las restantes metodologías dan origen a los mismos documentos que serán fundamentales para lograr una satisfactoria configuración e implantación del software educativo que se desea desarrollar.



1.6 Conclusiones del capítulo

En este capítulo se ofrecen los principales fundamentos teóricos relacionados con el objetivo de la investigación. Para ello se investigaron diferentes aspectos sobre los procesos de desarrollo de software educativos tales como: los conceptos que en la literatura especializada plantean diferentes autores, en relación a software educativo, así como las particularidades del desarrollo de dichos software en Cuba. Se estudiaron los distintos modelos de desarrollo de SWE, quedando evidenciada la necesidad de documentar dichos procesos.

Se identificaron las documentaciones generadas por las metodologías de desarrollo de software estudiadas en el capítulo. De manera que se proponen algunos documentos que no sufrirán grandes cambios manteniendo su estructura original. Ejemplo de ello son: Glosario de Términos, Especificación de Requisitos y de Casos de Uso, Documento Visión, Descripción de la Arquitectura del Software, Plan de Integración e Iteración, Plan de Pruebas, Manual de Instalación y de Usuario, Plan de Adiestramiento y por último el Plan de Implantación.

Otros documentos serán adaptados al SWE o creados exclusivamente para este tipo de software, con el objetivo de recopilar toda la información que se establece desde la concepción hasta la finalización del proceso de desarrollo. Haciendo mención a dichos documentos tenemos el Modelo Pedagógico, Guión de Contenidos, Mapa de Navegación, Pautas del Diseño Gráfico, Plan Audiovisual y Diseño de Pruebas Pedagógicas.



Capítulo#2: Desarrollo de la Propuesta

2.1 Introducción

Durante el desarrollo de este capítulo se realizará un análisis más detallado de las documentaciones que se generarán durante el proceso de desarrollo de software educativo, basándose en las propuestas abordadas anteriormente; además de otros documentos que serán asumidos para el proceso de desarrollo. Se tomará como punto de partida la metodología RUP y las investigaciones realizadas por ingenieros como Pere Marqués y Zulma Cataldi, obteniendo de cada una de ellas los documentos de salida que se crean en las fases de desarrollo que serán descritas brevemente.

2.2 Modelación Conceptual

El Modelo Conceptual es un conjunto de conceptos y de reglas destinados a representar de forma global los aspectos lógicos de los diferentes tipos de datos existentes en la realidad que está siendo analizada; ha de permitir reflejar el contenido semántico de los datos existentes en el sistema, pero no sus propiedades que respondan a características de tipo físico (modo de almacenamiento, caminos de acceso, etc.)

Esta fase de desarrollo, por la que atravesará el SWE, estará representada por un diagrama que ilustra una serie de relaciones entre ciertos factores que se creen, impactan o conducen a una condición de interés. Un buen Modelo Conceptual muestra un cuadro de la situación del software a desarrollar, mostrando supuestos vínculos entre los factores que afectan a la condición de interés, así como las principales amenazas directas e indirectas que afectan a dicha condición. Además presenta sólo factores relevantes, en este caso del SWE, y está basado en datos e información concreta. Además debe estar regido por las siguientes propiedades: (Margoluis, 1998)

- Completo
- Mínimo
- Fácil de comprender
- Fácil de formular
- Formal
- Explícito en todas sus restricciones



2.2.1 Documentación Necesaria

Durante esta primera fase de desarrollo se generarán documentos que a continuación se detallan con el objetivo de sustentar el proceso de desarrollo SWE.

❑ Modelo pedagógico

No es más que el contenido de la enseñanza, el desarrollo del estudiante y la característica de la práctica del docente. Este modelo no sólo impone como exigencia establecer el modo en que se han de entregar dichos contenidos a los estudiantes, sino también, cuál será la participación de ellos en la búsqueda o recepción de la información. Es el instrumento de la investigación de carácter teórico creado para reproducir idealmente el proceso enseñanza - aprendizaje.

Este documento que se propone para este primer subproceso del SWE (Ver Anexo#3) estará contenido por el diseño instruccional de los productos educativos; además de reflejar los contenidos educativos organizados por su prioridad y complejidad, pues cada secuencia de contenidos estará diseñada en función de unos objetivos a lograr por el alumno.

❑ Guión de contenidos

Es el encargado de presentar de la forma más esquemática posible aquello que se quiere cambiar. Indica el material textual que se va a utilizar en las diferentes etapas por las que atravesará el software educativo, y la manera en la que se va relacionando mediante una jerarquización conceptual que deberá transmitirse en forma muy clara en el guión. Será el resultado del proceso de obtención de los contenidos con respecto a las definiciones en el Modelo Pedagógico.

Para la realización de la propuesta del documento Guión de Contenidos (Ver Anexo#9) el desarrollo del presente trabajo se basó en algunos de los aspectos analizados en los modelos evaluativos y en la metodología para la elaboración de un SWE propuesto por Pere Marqués.

El documento estará conformado por una lista detallada de elementos que constituirán el proyecto educativo/instruccional, detallando el funcionamiento de cada uno de los contenidos en cada una de las pantallas del proyecto. Dichos contenidos estarán estructurados modularmente y las actividades organizadas de manera secuencial. Además contendrá una estrategia educativa



conformada por objetivos, contenidos, destinatarios y operaciones mentales que tienen que desarrollar los alumnos. Se describirán los recursos audiovisuales listados en el SW a través de un identificador y la descripción del mismo.

❑ **Mapa de Navegación**

Es un esquema detallado de las opciones que va a tener el usuario en cada parte del sistema; es decir representa la arquitectura física que forman las páginas y los enlaces hipertextuales, mostrando una estructura de los nodos para agilizar la navegación por la Web. Con los mapas de navegación de podrá hacer un bosquejo de las conexiones de las distintas áreas de contenido.

En el documento Mapa de Navegación (Ver Anexo#8) que se propone, apoyándose en investigaciones realizadas por el ingeniero Pere Marqués, se analizarán aspectos como: la navegación de la aplicación pedagógica; es decir, la manera en que estarán distribuidos cada unos de los contenidos en dicha aplicación. Además se dará un entorno pedagógico y una breve descripción de las actividades que se generarán en la aplicación.

❑ **Glosario de términos**

El documento Glosario de Términos (Ver Anexo#6) es un artefacto tomado de RUP, tal y como esta metodología la genera. Almacena los términos y definiciones de los términos a hacer utilizados durante la realización del proyecto; (diccionario de datos) que deben ser comprendidos por los participantes de tal manera que haya una buena comunicación y evitar interpretaciones dispares o ambiguas de los términos del dominio del problema. En su forma más simple, es una lista de los términos relevantes y sus definiciones.

2.3 Gestión de Requisitos

La Gestión de Requisitos es el conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento. Es decir consiste en gestionar los cambios de los requisitos, las relaciones entre ellos, las dependencias entre la especificación de requisitos y otros documentos producidos por el proceso de desarrollo de software. De esta forma se asegura la consistencia entre los requisitos y el sistema construido. (Ian Sommerville, 1997)



Consume grandes cantidades de tiempo y esfuerzo, abarcando todo el ciclo de vida del producto. Es una actividad necesaria porque: (Ingeniería de Requisitos, 2007)

- Los requisitos son volátiles:
 - Mutantes o cambiantes: Sufren ligeras variaciones.
 - Emergentes: Aparecen al ir investigando o analizando el sistema en profundidad.
 - Colaterales: Surgen como efecto de la inclusión de otros requisitos.
 - Por compatibilidad: Se añaden para adaptar el sistema a su entorno, debido a que el entorno físico cambia, o sea, trasladar un sistema de un entorno a otro siempre requiere modificaciones. El entorno organizacional también cambia, las políticas cambian, se producen cambios en las reglas y en los procesos del negocio, provocando cambios en el sistema.
- La propia existencia del sistema va a generar nuevos requisitos por parte de los usuarios.

Con este proceso será necesario no solo definir y detallar correcta y claramente los requisitos, sino que será preciso darle un seguimiento durante todo el ciclo de vida del proyecto hasta su implementación y mantener un control apropiado de los cambios. Igualmente, de dichos requerimientos, definirá su funcionalidad, restricciones y características particulares del futuro sistema.

2.3.1 Documentación Necesaria

Durante todo el proceso de requisitos se generan documentos que darán apoyo a la solución del software y mantendrán recogida toda la información que se procese durante su desarrollo:

❑ Especificación de Requisitos

Captura otros requisitos, información y restricciones que no se recogen fácilmente en los casos de uso o el Glosario que comprende los atributos o requisitos de calidad de todo el sistema. Estos comprenden:

- Facilidad de uso, fiabilidad, rendimiento y soporte.
- Informes.
- Restricciones de Software y Hardware.
- Restricciones de Desarrollo.



- Asuntos de internacionalización.
- Documentación (usuario, instalación, administración) y Ayuda.
- Licencia y asuntos legales.
- Empaquetado.
- Estándares (técnicos, de seguridad y de calidad).
- Factores del Entorno Físico.
- Factores Operacionales.

La elaboración de este documento (ver Anexo #5) se fundamentó en estudios realizados por Pere Marqués y en las documentaciones que propone la metodología RUP para esta fase de desarrollo. En el mismo se detallarán todos los requisitos del sistema, separándolos en funcionales y no funcionales. Los **requerimientos funcionales** no son más que capacidades o condiciones que el sistema debe cumplir; es decir, no alteran la funcionalidad del producto; a diferencia de los **requerimientos no funcionales**, que no son más que propiedades o cualidades que el producto debe tener. Dígase propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

❑ Documento Visión

La creación de esta planilla fue implantada por la metodología RUP, haciéndole algunas adaptaciones para el software educativo, con el objetivo de que los usuarios finales logren un mejor entendimiento del contenido o información a tratar en la aplicación.

Este documento (Ver Anexo#4) sirve para comunicar de manera concisa las grandes ideas acerca de por qué se propuso el proyecto, cuáles son los problemas, quiénes son las personas involucradas, qué necesitan, y cuál podría ser la apariencia de la solución propuesta. Revisa si se está solucionando el mismo problema y si se está resolviendo el problema correcto.

Dar una visión del proyecto es uno de los primeros pasos durante todo proceso de desarrollo, concediendo una perspectiva clara y general de todo el contenido a tratar en el producto audiovisual y las necesidades que se van a cubrir con el proyecto, enfatizando en aspectos como: características básicas del software audiovisual, sus funcionalidades y un pequeño resumen del



análisis didáctico. Se encargará de precisar los pormenores de cómo el sistema satisface las necesidades para los implicados y usuarios finales.

2.4 Análisis y Diseño del Sistema

Antes de profundizar en el Análisis y el Diseño, es válido destacar que para la modelación de las clases en el proceso de desarrollo del software educativo, se tuvo en cuenta la propuesta que hace el ingeniero Febe Ángel Ciudad Ricardo: ApEM-L, que no es más que una nueva solución para la modelación de aplicaciones educativas. Este nuevo lenguaje de modelado estará agrupado por las siguientes áreas conceptuales: estructura lógica, comportamiento dinámico y gestión del modelo; además se mostrarán las vistas que modelarán cada una de las construcciones que representan un aspecto del sistema: vista estática, vista de arquitectura, vista de comportamiento y vista de presentación. (Ricardo, 2007)

Durante la fase de desarrollo del **Análisis** se podrá estructurar los requerimientos tratados anteriormente de manera que nos facilite su comprensión, preparación, modificación, y en general su mantenimiento; es decir se obtendrá una visión del problema, preocupándose por ver QUÉ hace, de manera que solamente interesan los requisitos funcionales. El equipo del proyecto profundizará en el dominio de la aplicación con el objetivo de lograr una mayor comprensión del problema para modelar la solución.

Para ello se somete al sistema a una arquitectura del software, la cual según Pressman, alude a “la estructura global del software y a las formas en que la estructura proporciona la integridad conceptual de un sistema”. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes. Sin embargo, en un sentido más amplio, los “componentes” se pueden generalizar para presentar los elementos principales del sistema y sus interacciones. (Pressman, 2002)

En el caso del subproceso de **Diseño**, el cual no es más que el refinamiento del Análisis, se tendrá en cuenta los requisitos no funcionales obtenidos después de un previo estudio del producto educativo a desarrollar; es decir, es el CÓMO cumple el sistema sus objetivos.

Esta fase debe ser suficiente para que la aplicación pueda ser implementada sin imprecisiones, o sea, es el proceso donde se aplicarán técnicas y principios con el fin de lograr un sistema o aplicación pedagógica y didáctica con suficientes detalles, como para permitir una correcta interpretación y realización física.



El proceso de diseño es la piedra angular para la obtención de un producto coherente que satisfaga los requisitos de software. Desde el punto de vista técnico comprende cuatro tipos de actividades: diseño de datos, arquitectónico, procedimental y diseño de interfaces y desde el punto de vista del proyecto evoluciona desde un diseño preliminar al diseño detallado. (Cataldi, 2000)

Otro de los aspectos fundamentales que debe contemplarse en el diseño de un SWE es que puede posibilitar al estudiante, el descubrir para qué estudia los diferentes contenidos y cómo a través del análisis de los mismos, se pueden resolver disímiles problemas.

2.4.1 Documentación necesaria para ambos procesos

Los documentos que se generan en el Análisis por trabajadores como el arquitecto, ingeniero de casos de uso, y de componentes, estarán muy ligados a los que se crearán en el Diseño. Para generar estos informes se tomó como referencia las documentaciones que genera RUP durante el desarrollo de estos dos procesos analizados.

❑ Especificación de Casos de Uso

Este documento (ver Anexo #11) es el responsable de describir cómo se lleva a cabo y se ejecuta un caso de uso determinado con respecto a las clases de análisis y de sus objetos en interacción. Además define el Modelo de casos de uso dando una descripción de los mismos; para ello se propone que se utilice el lenguaje ApEM-L. (Ricardo, 2007)

❑ Descripción para la Arquitectura de Software

Este documento (ver Anexo #7) es el encargado de mostrar los artefactos relevantes para la arquitectura del software, dando una vista arquitectónica del Modelo de Análisis. Además en esta planilla se especificarán las ideas principales del diseño y proporcionará una descripción entendible de la arquitectura del sistema software, sirviendo como medio de comunicación entre el arquitecto de software y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicamente significativas que se han tomado en el proyecto.

Contiene varias vistas que muestran aspectos distintos del sistema; para ello resulta necesario utilizar la propuesta de (Ricardo, 2007) para modelar el sistema de vistas arquitectónicas.



2.5 Diseño Gráfico

Con el Diseño Gráfico, como subproceso del desarrollo del software educativo se pretende concebir, programar, proyectar y realizar comunicaciones visuales, producidas en general por medios industriales y destinados a transmitir mensajes educativos al usuario final; además de darle un estilo personal al software. Para ello es necesario tener en cuenta el Modelo Pedagógico y la Especificación de Requisitos para dar cumplimiento a las necesidades que pide el software.

2.5.1 Documentación Necesaria

□ Pautas del Diseño Gráfico

Durante el llenado de esta planilla (ver Anexo#19) se abordarán aspectos específicos del diseño gráfico, tales como: diagramación general, dimensiones, estilo de menú, botones e imágenes, pautas para el diseño de banners, pautas tipográficas, etc. Cada uno de estos aspectos serán los encargados de darle vida, estilo y calidad al SWE.

2.6 Gestión Audiovisual

El objetivo de ese proceso es gestionar los recursos audiovisuales que fueron identificados durante la etapa de Modelación Conceptual con la calidad técnica-estética requerida.

Con la Gestión Audiovisual se puede obtener una sugestiva presentación, así como una estructura de las pantallas, composición, tipografía, colores, disposición de los elementos multimedia, estética, etc. Para ello debe tener:

- Una presentación atractiva y correcta. La resolución óptima para su visualización debe ser de (800x600...)
- Un diseño claro y atractivo de las pantallas, sin exceso de texto, destacando lo importante.
- Una calidad técnica y estética en sus elementos: títulos, barras de estado, frames, menús, barras de navegación, etc.

Este proceso servirá de apoyo al SWE facilitando las explicaciones del profesorado y el desarrollo de actividades participativas en gran grupo.



2.6.1 Documentación Necesaria

❑ Plan Audiovisual

Este documento (ver Anexo#20) se encargará de recoger los recursos educativos empleando un código identificador, dando paso a la caracterización de cada una de las medias, enunciando de éstas su localización física.

2.7 Implementación

Este subproceso se encarga de explicar cómo desarrollar, organizar, realizar pruebas de unidad e integrar los componentes implementados basándose en las especificaciones de diseño; es decir trata de convertir los elementos del diseño en elementos de implementación, dichos elementos son códigos fuentes, ejecutables, binarios, entre otros. Otra parte de esta disciplina son las pruebas de unidad, las cuales se limitan a los componentes de software implementados. De esta disciplina se obtiene un sistema ejecutable estable, constituido de los resultados producidos por los programadores individuales.

La implementación tiene como finalidad:

- Definir la organización del código, en términos de los subsistemas de implementación organizados en capas.
- Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros).
- Probar y desarrollar componentes como unidades.
- Integrar los resultados producidos por los implementadores individuales (o equipos) en un sistema ejecutable.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.



2.7.1 Documentación Necesaria

Para realizar la propuesta de documentación, la investigación se ha basado en los artefactos que propone la metodología RUP, artefactos que darán como salida documentos que serán necesarios para un mejor entendimiento del trabajo realizado en esta fase de desarrollo.

❑ Plan de Integración

Este documento (ver Anexo #12) muestra un plan detallado de la integración dentro de una iteración. Se encargará de definir el orden en que los componentes deben implementarse, qué construcciones deben crearse cuando se integra el sistema, y cómo deben valorarse.

❑ Plan de Iteración

El documento Plan de iteración (ver Anexo #13) tiene como objetivo definir detalladamente para cada una de las iteraciones a realizarse un conjunto de tareas, actividades y recursos. Por tal motivo existirá para cada iteración del ciclo de vida del proyecto un artefacto de este tipo. Para cada iteración existe una serie de objetivos los cuales son usados como referencia de evaluación para determinar diferentes aspectos, como grado de terminación de una determinada función, rendimiento, niveles de calidad, etc.

Para cada plan de iteración es necesario detallar la programación estimada para la iteración, los recursos a emplear, los casos de uso y escenarios que van ser tomados en cuenta y finalmente se deben establecer los criterios de evaluación que se van a tener para la iteración. Es recomendable para las iteraciones emplear herramientas para la planeación de proyectos con el fin de hacer más fácil y organizada esta tarea. De ser empleada cualquier herramienta sus resultados debe ser reflejados en el plan de iteración.

Para poder definir una iteración es necesario tomar en cuenta:

1. La planificación del proyecto.



2. El estado actual en el que se encuentra el proyecto (proyecto dentro de los tiempos estipulados, proyecto retrasado con respecto al tiempo estipulado, un gran número de problemas encontrados, etc).
3. Los elementos a ser implementados. La lista de casos de uso y de escenarios que deben ser cumplidos al final de la iteración.
4. La lista de los cambios que deben ser incorporados (corrección de errores, cambios de requerimientos).
5. Los riesgos que se pueden correr en la iteración.

2.8 Prueba

El principal objetivo de esta disciplina es de evaluar la calidad del producto que se está desarrollando a través de las diferentes fases por las cuales este pasa, mediante la aplicación de pruebas concretas para validar que las suposiciones hechas en el diseño y los requerimientos se estén cumpliendo satisfactoriamente, esto quiere decir que se verifica que el producto funcione como se diseñó y que los requerimientos son satisfechos cabalmente. Esta disciplina brinda soporte para encontrar y documentar defectos en la calidad del sistema a las otras disciplinas. Debe estar presente en todo el ciclo de vida del desarrollo del sistema para ir refinándolo y no al final del mismo. Como principales objetivos presenta:

1. Encontrar y documentar defectos en la calidad del software.
2. Notificar la calidad percibida del software.
3. Proveer un medio de validación para las suposiciones hechas en el diseño y especificaciones de requerimientos por medio de demostraciones concretas.
4. Validar las funciones del producto de software según lo diseñado.
5. Validar que los requerimientos fueron implementados apropiadamente.

El desarrollo de esta disciplina consistirá en planificar qué es lo que hay que probar, diseñar cómo se va a llevar a cabo la prueba, implementar lo necesario para llevarlas a cabo, ejecutarlas en los niveles necesarios y obtener los resultados, de forma que la información obtenida sirva para ir refinando el producto a desarrollar.



El papel de las pruebas no es asegurar la calidad, pero sí evaluarla, y proporcionar una realimentación a tiempo, de forma que los aspectos de calidad puedan resolverse de manera efectiva en tiempo y costo.

Los principales aspectos a ser evaluados en un producto software son la Funcionalidad (hace lo que debe), la Fiabilidad (resistente a fallos), y el Rendimiento (lleva a cabo su trabajo de manera efectiva). Las pruebas pueden hacerse a diferentes niveles dependiendo del objetivo de los mismos, entre algunos tenemos: Pruebas de unidad (se prueban las unidades mínimas por separado, y normalmente se hace durante la implementación misma), de integración (varias unidades juntas), de sistema (sobre la aplicación o sistema completo) y de aceptación (realizado sobre el sistema global por los usuarios o terceros).

2.8.1 Documentación Necesaria

❑ Plan de Pruebas

El documento Plan de Pruebas (ver Anexo#14) será el encargado de coleccionar los casos de prueba y procedimientos de prueba. Este artefacto incluye el propósito de las pruebas, qué elemento se va a probar, las herramientas a utilizar y con qué recursos. Al tener el resultado de las pruebas se puede comparar lo obtenido con lo esperado. Además en esta planilla se reflejarán las características de hardware y software que serán empleados para realizar el conjunto de las pruebas al sistema. Otro de los aspectos que recogerá este documento es un listado de ideas de pruebas, ideas que potencialmente serán las pruebas más útiles a realizar.

❑ Diseño de Pruebas Pedagógicas

El Diseño de Pruebas Pedagógicas es un proceso que está basado en la evaluación del SWE para determinar características, fortalezas, debilidades y aplicabilidad en las distintas áreas de enseñanza-aprendizaje del software.

Esta evaluación tiene como finalidad orientar un uso pedagógicamente adecuado, para lo cual se ha centrado en dos momentos del desarrollo y uso de este tipo de materiales: (González, 1996)

1. Durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa.
2. Durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen.



Según el ingeniero Castañón, se llega a la conclusión de que el primer momento es el más frecuente debido a que los diseñadores de software mantienen como mecanismo necesario la evaluación del proceso de diseño y producción; es decir evalúan los aspectos computacionales y algunos referidos a la relación hombre-máquina y a la facilidad o funcionalidad de uso del programa.

En el documento Diseño de Pruebas Pedagógicas (Ver Anexo#10) se tratarán aspectos como la usabilidad del programa, contenido científico, socio-cultural e ideológico, pedagógico, entre otros que serán explicados durante su desarrollo.

Muchos son los modelos y pautas que se han desarrollado para la evaluación del SWE. De acuerdo a ello se consideran las propuestas realizadas por autores o grupos de investigación, tal es el caso de (Squires, 1997,2007) con el *Paradigma de las interacciones de perspectivas aplicado al software educativo*, (P.Clarke, 1997) con el *Modelo de Evaluación de Software Educativo*, (B.Poole, 1999) con la *Lista de control para la evaluación de software educativo*, (Marqués, 2004) y (P.Marqués, 2005) con una *Ficha de catalogación y evaluación multimedia*, entre otros.

2.9 Implantación

Esta disciplina tiene como objetivo distribuir e instalar con éxito el sistema elaborado por el equipo de desarrollo y asegurar la disponibilidad del producto para los usuarios finales. Dentro de sus principales objetivos tenemos:

1. Probar el producto en su entorno de ejecución final.
2. Proveer asistencia y ayuda a los usuarios.
3. Migrar el software existente.
4. Instalar el software.
5. Formar a los usuarios y al cuerpo encargado de distribuir el sistema.

Se lleva a cabo con mayor intensidad en la fase de transición, debido a que su propósito es asegurar una aprobación y adaptación sin que existan dificultades del software por parte del usuario. Esta disciplina debe iniciar en fases anteriores, para preparar el camino, sobre todo con actividades relacionadas a la planificación, pero también con la elaboración del manual de usuario y tutoriales. Debido al extenso nivel



de aplicaciones que se pueden obtener y las diversas características de los productos necesarios para esta disciplina, se pueden obtener grandes variaciones dependiendo del tipo de sistema a desarrollar. El objeto clave es una distribución del producto.

Aunque el sistema esté bien diseñado y desarrollado correctamente su éxito dependerá de su implantación y ejecución por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

2.9.1 Documentación Necesaria

❑ Manual de Instalación

El Manual de Instalación (ver Anexo#15) es un artefacto que refleja los lineamientos que hay que seguir para instalar el sistema. Contiene información sobre la infraestructura de instalación e instrucciones para la instalación y actualización del software.

❑ Manual de Usuario

Este documento (ver Anexo#16) provee una ayuda a las personas que manipularán directamente el producto, acerca del uso que le debe dar al sistema. Dicho artefacto debe ser discutido y aprobado por el cliente.

Elaborar el manual de usuario durante las primeras iteraciones del proyecto permitirá al equipo de probadores conocer el sistema antes de que comiencen las pruebas. Adicionalmente provee los mecanismos básicos para elaborar los planes de pruebas y los casos de pruebas, y permite la elaboración de sistemas automatizados para las pruebas.

Según el tipo de sistema se define el comienzo del desarrollo del Manual de Usuario. Sistemas con interfaces complejas o con mucha interacción requerirán versiones tempranas del manual de usuario así como de prototipos de interfaces. Sistemas con poca interacción probablemente no requieran que la documentación del usuario se elabore muy temprano.



❑ **Plan de Adiestramiento**

Muestra el plan detallado de adiestramiento. El propósito de este plan (ver Anexo#17) es que las personas que vayan a utilizar el sistema, se capaciten para su utilización evitando que el mismo sea mal utilizado. Dicho artefacto debe ser discutido y aprobado por el cliente.

Este artefacto está compuesto por secciones para indicar el programa de entrenamiento o cursos que serán impartidos a los usuarios finales para enseñarles el uso, operación y mantenimiento del sistema.

❑ **Plan de Implantación**

El objetivo principal de este artefacto es asegurar que el sistema llegue satisfactoriamente al conjunto de usuarios para el cual fue destinado. Este documento debe definir un conjunto de tareas que defina una transición sencilla para el cliente, para ello se debe minimizar el impacto que la implantación del sistema pueda llegar a causar en el personal del cliente, los sistemas de producción existentes y en todas las rutinas del negocio.

Esta planilla (ver Anexo#18) describe el conjunto de tareas necesarias para poner en funcionamiento el sistema en las instalaciones de los usuarios. Las actividades descritas en este documento abarcan temas referentes a la instalación del nuevo sistema, instrucciones específicas sobre la sustitución de antiguos sistemas, compatibilidad del sistema, y estrategias de migración y adaptación al nuevo sistema. Adicionalmente este artefacto describe en detalle las actividades correspondientes a la entrega del producto, el cronograma de actividades, personal responsable, los recursos y fuentes necesarias para el funcionamiento del nuevo sistema, plan de adiestramiento, notas de seguridad, de procedimientos operacionales específicos, entre otros.



2.10 Conclusiones del capítulo

Durante el desarrollo del presente capítulo se definió brevemente cada uno de los subprocesos por los que atravesará el software educativo, haciendo énfasis en las documentaciones generadas por las distintas metodologías estudiadas en el capítulo anterior. Muchos de los documentos presentados estuvieron sujetos a cambios para ser adaptados al SWE, y otros fueron totalmente creados para fundamentar las fases de desarrollo del sistema educativo.

Los mencionados documentos algunos servirán para mantener una correcta estructura y planificación durante la evolución del proyecto, y de este forma evitar que se ponga en riesgo la calidad del producto final; y otros serán de ayuda para mantener informado al usuario del contenido del software y lograr un correcto uso de sus funcionalidades.



Capítulo#3: Validación de la Propuesta

3.1 Introducción

Durante el desarrollo del presente capítulo se analizarán los resultados mediante una selección de expertos, que darán una evaluación de la propuesta planteada en el capítulo anterior. Para ello se realizará un estudio detallado del Método Experto conformado por un grupo de expertos que serán los encargados de exponer sus ideas y finalmente redactarán un informe en el que se indican cuáles son, en su opinión, las posibles alternativas que se tendrán en el futuro.

3.2. Método de Experto

El Método Experto es el que permite, según las decisiones tomadas por expertos, tomar o rechazar determinada propuesta. Este grupo de expertos, seleccionados por su amplio conocimiento y desarrollo en materia de software educativo, serán los encargados de analizar las convergencias de opiniones en torno al problema que aborda la investigación.

3.2.1 Aplicación del Método Experto

Para efectuar la validación de la propuesta por el método experto se hace necesario definir un conjunto de pasos que serán explicados a continuación:

Paso 1: Conformar criterios de evaluación y organizarlos por grupos.

- Grupo No. 1: Criterios de Valor científico
 - ✓ Valor científico de la investigación.
 - ✓ Calidad de la investigación desarrollada.
 - ✓ Novedad científica de la investigación.
- Grupo No. 2: Criterios de implantación
 - ✓ Necesidad real de implantación de la propuesta.
 - ✓ Posibilidad de ser aplicada la propuesta.
 - ✓ Integración de la propuesta con el proceso de desarrollo de SWE en la UCI.



- Grupo No.3: Criterios de adaptación
 - ✓ Adaptabilidad de la propuesta al proceso de desarrollo de SWE en la UCI.
 - ✓ Integración de la propuesta con la metodología de desarrollo utilizada por los proyectos.
 - ✓ Integración con el entorno de producción donde se aplique.
- Grupo No.4: Criterios de impacto
 - ✓ Repercusión de la propuesta en el proceso de desarrollo de SWE en la UCI
 - ✓ Organización en el proceso de documentación de la investigación
 - ✓ Fluidez de los procesos de desarrollo de SWE en la UCI como el uso de la propuesta

Paso 2: A cada grupo de criterios se le asignará un peso relativo, sumando un total de 100.

Grupo No.1: Criterios de Valor científico.....	25
Grupo No. 2: Criterios de implantación.....	25
Grupo No.3: Criterios de adaptación.....	20
Grupo No.4: Criterios de impacto.....	30

Paso 3: Seleccionar un comité de expertos conformado por un mínimo de 7 expertos. Esta elección estuvo basada en la especialidad, grado científico y currículum de cada uno.

Paso 4: Se entregará a cada experto la propuesta para que sea estudiada, y evaluada a través de dos modelos, en el primero se valorará el peso relativo de cada criterio (Ver Anexo#21) y en el segundo se realizará una evaluación cuantitativa de dichos criterios en una escala del 1 – 5 (Ver Anexo#22). Seguidamente se debe registrar una apreciación cualitativa de la propuesta de *excelente, bueno, aceptable, cuestionable y malo*; así como consideraciones en relación a dicha propuesta.

Paso 5: Calcular por cada criterio el peso promedio, partiendo de los pesos dados por los expertos. El peso promedio por cada criterio se muestra en la siguiente tabla:



Tabla 8: Peso Promedio/Criterio

G	Exp/Crit	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	Exp. P
25	C1	9	6	10	9	9	8	9	60	8.571429
	C2	8	10	10	9	8	8	8	61	8.714286
	C3	8	9	5	7	8	9	8	54	7.714286
25	C4	10	8	10	10	10	8	8	64	9.142857
	C5	8	9	5	8	7	9	7	53	7.571429
	C6	7	8	10	7	8	8	10	58	8.285714
20	C7	8	10	7	5	6	7	7	50	7.142857
	C8	6	4	6	7	7	6	6	42	6
	C9	6	6	7	8	7	7	7	48	6.857143
30	C10	11	8	8	12	10	9	9	67	9.571429
	C11	10	9	11	10	9	10	11	70	10
	C12	9	13	11	8	11	11	10	73	10.428571
Totales		100	100	100	100	100	100	100	700	100.000

Paso 6: Determinar la consistencia en el trabajo de los expertos, utilizando el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (χ^2). Dado C el número total de criterios a evaluarse, y E el número de expertos involucrados en la evaluación, se realiza el siguiente procedimiento para determinar la consistencia en el trabajo de los expertos:

- Calcular para cada criterio: ΣE , que representa la sumatoria del peso dado por los expertos.
- Determinar el valor de Exp. P: puntuación promedio de cada criterio.
- Calcular el peso medio de cada criterio $M\Sigma E = \Sigma E / Cn$.
- Hallar el valor de ΔC , que representa la diferencia entre ΣE y $M\Sigma E$.



- Determinar la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión S, dada por la expresión: $S = \Sigma (\Sigma E - \Sigma \Sigma E / C)^2$.
- Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall W, dado por la expresión: $W = S / E^2 (C^3 - C) / 12$.
- Calcular el Chi cuadrado real a partir del valor del coeficiente de Kendall teniendo en cuenta la siguiente expresión: $X^2 = E (C-1) W$.

A continuación se muestran los datos obtenidos luego de realizar los pasos anteriores.

Tabla 9: Consistencia en Trabajo de Expertos.

E/C	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	Exp. P	ΔC	ΔC ²
C ₁	9	6	10	9	9	8	9	60	8.571429	1.67	2.7890
C ₂	8	10	10	9	8	8	8	61	8.714286	2.67	7.1290
C ₃	8	9	5	7	8	9	8	54	7.714286	4.33	18.7490
C ₄	10	8	10	10	10	8	8	64	9.142857	5.67	32.1490
C ₅	8	9	5	8	7	9	7	53	7.571429	5.33	28.4090
C ₆	7	8	10	7	8	8	10	58	8.285714	0.33	0.1090
C ₇	8	10	7	5	6	7	7	50	7.142857	8.33	69.3890
C ₈	6	4	6	7	7	6	6	42	6	16.33	266.6690
C ₉	6	6	7	8	7	7	7	48	6.857143	10.33	106.7090
C ₁₀	11	8	8	12	10	9	9	67	9.571429	8.67	75.1690
C ₁₁	10	9	11	10	9	10	11	70	10	11.67	136.1890



C₁₂	9	13	11	8	11	11	10	73	10.428571	14.67	215.2090
Totales	100	100	100	100	100	100	100	700	100.000	90	958.668
MΣE	58.33										
W	0.1368										
X²_{real}	10.5336										

Posteriormente, se compara el X^2 real, con el valor del dato estadístico, siendo $\alpha=0.01$, y $C = 12$ y debe cumplirse que $X^2 < X^2(\alpha, c-1)$ para que el trabajo realizado por los expertos sea valorado de consistente. $X^2(\alpha, c-1) = X^2(0.01, 11) = \mathbf{24.7250}$.

Por tanto $10.53 < 24.73$, quedando demostrada la consistencia en el trabajo realizado por los expertos.

Paso 7: Calcular $P \times c$ donde “P” es el peso relativo de cada criterio y c es la calificación promedio dada por los expertos.

P: peso promedio de cada criterio.

$$P = (\text{Exp. P})/100$$

c: calificación promedio de cada criterio concebida por los expertos.

Tabla 10: Cálculo de P*c.

Criterios	Calificación (c)					P	P * c
	1	2	3	4	5		
C₁				X		0.085	0.340
C₂				X		0.087	0.348
C₃				X		0.077	0.308
C₄					X	0.091	0.455



C ₅				X		0.075	0.300
C ₆				X		0.082	0.328
C ₇				X		0.071	0.284
C ₈				X		0.06	0.240
C ₉				X		0.068	0.272
C ₁₀				X		0.095	0.380
C ₁₁					X	0.100	0.500
C ₁₂				X		0.104	0.416

Paso 8: Se calcula el Índice de Aceptación (IA) de la propuesta.

Se calcula a partir de la siguiente fórmula teniendo que $\sum (P * c) = 4.171$

$$IA = (P * c) / 5$$

Si $(P * c) / 5 = 4.171$ entonces $IA = 4.171 / 5$ y se obtiene **IA = 0.8342**.

Paso 9: Determinar la probabilidad de éxito de la propuesta a partir de los rangos predefinidos del índice de aceptación:

IA > 0,7.....Existe alta probabilidad de éxito.

0,7 > IA > 0,5.....Existe probabilidad media de éxito.

0,5 > IA > 0,3.....Existe baja probabilidad de éxito.

0,3 > IA.....No existe probabilidad ninguna de éxito.

Dado el resultado de IA es igual a **0.8342** entonces se puede concluir que la probabilidad de éxito es: ALTA; y se establece por los expertos una valoración final de: BUENO.



3.3. Conclusiones del Capítulo

Una vez utilizado el Método de Expertos y por ende, validadas las actividades y tareas, se llega a la conclusión de que todo el trabajo investigativo puede ser aplicado al desarrollo del software educativo. De esta manera se demuestra la factibilidad y utilidad de esta propuesta para lograr una mejor calidad del proceso de desarrollo de software educativo.



Conclusiones Generales

Basado en todo el proceso investigativo del presente trabajo de diploma, para el desarrollo del Software Educativo se arribaron a las siguientes conclusiones generales:

- Se realizó una investigación detallada de las documentaciones existentes en cada una de las metodologías analizadas en el transcurso del trabajo logrando una fundamentación consistente para la realización de la propuesta.
- Sobre la base del proceso de desarrollo de software educativo, definido en investigaciones realizadas en la UCI por la Ing. Yaillet Martínez Pérez, se propuso una documentación para cada uno de los subprocesos descritos para este tipo de software, alcanzando una centralización y organización de la información almacenada.
- Se llevó a cabo la evaluación de los criterios reconocidos por el Método de Expertos, evidenciando la aplicabilidad de la propuesta mediante un índice de aceptación con una probabilidad de éxito alta.



Recomendaciones

Una vez cumplido cada uno de los objetivos trazados en la investigación se recomienda:

- Desarrollar una aplicación Web de ayuda que propicie el acceso a los proyectos de la UCI a las documentaciones propuestas para el proceso de desarrollo de software educativo.
- Que la propuesta de documentación sea utilizada por los proyectos de SWE en la UCI u otros centros productores para validarla en la práctica.
- Definir las métricas para evaluar la calidad de la documentación generada.



Referencias Bibliográficas

- ✚ **B.Poole. 1999.** *Tecnología Educativa.* s.l., España : Mc Graw Hill, 1999. Educar para la sociedad de la comunicación y del conocimiento.
- ✚ **Boehm, B. 1988.** *A spiral model of software development and enhancement.* *Computer 1988 IEEE.* 1988. págs. 61-72.
- ✚ **Cataldi, Zulma. 2000.** Metodología de diseño, desarrollo y evaluación de software educativo. [En línea] 2000. [Citado el: 04 de 05 de 2009.] Facultad de Informática. Universidad Nacional de La Plata. <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>.
- ✚ **DoD. 1994.** *Software Development and Documentation.* Military Standard 498, s.l. : Department of Defense of the United States of America, 1994.
- ✚ **Gagné, R. 1987.** *Instructional technology.* s.l. : Foundations Hillsdale, 1987. Lawrence Erlbaum Associates.
- ✚ **Gálvis, Alvaro. 1992.** *Ingeniería de software educativo.* Bogotá : Ediciones Uniandes, 1992.
- ✚ **González, Miguel Angel Castañón. 1996.** Evaluación de Software Educativo: Orientaciones para su uso pedagógico. Medellín, Colombia : Proyectos Conexiones, 1996. Coordinador Area de Evaluación. Universidad EAFIT.
- ✚ **Ian Sommerville, Pete Sawyer. 1997.** *Requirements Engineering: A Good Practice Guide.* 1era edición. New York : Jhon Wiley & Sons, Inc. New York, NY, USA, 1997. pág. 404.
- ✚ *Ingeniería de Requisitos.* **UCI. 2007.** La Habana : s.n., 2007. pág. 13.
- ✚ **Maddison, R.N. 1983.** *Information System Methodologies.* 1983. Wiley Henden.
- ✚ **Margoluis, Nick Salafsky y Richard. 1998.** *measures of Success. Designing, Managing, and Monitoring Conservation and Development Projects.* 1998. pág. 198.



- ✚ **Marqués, Graell Pére. 1995.** Metodología para la elaboración de Software Educativo. Guía de uso y metodología de diseño. [En línea] 1995. [Citado el: 24 de 05 de 2009.] www.xtec.es/-pmarques/edusof.htm , www.doe.d5.ub.es.
- ✚ **Marqués, Graell Pére. 1998.** Programas didácticos: diseño y evaluación. *Universidad Autónoma de Barcelos*. [En línea] 1998. [Citado el: 15 de 3 de 2009.] www.doe.d5.ub.es/te.
- ✚ **Marqués, P. 2004.** Plantilla para la Catalogación y Evaluación Multimedia. [En línea] 2004. [Citado el: 08 de 03 de 2009.] <http://dewey.uab.es/pmarques/evalua.htm>.
- ✚ **P.Clarke, M.Pete, A.Naidoo. 1997.** Evaluation of Software and its effect on learning. [En línea] abril de 1997. [Citado el: 10 de 05 de 2009.] Kwa-Zulu Concept Burger: HTML mind/model of evaluation of educational software. <http://www.und.ac.za/users/clarke/kzb/dimeval.htm>.
- ✚ **P.Marqués. 2005.** Entorno formativo multimedia: elementos, plantilla de evaluación/criterios de calidad. [En línea] 2005. [Citado el: 08 de 05 de 2009.] <http://dewey.uab.es/pmarques/calidad.htm>.
- ✚ **Pérez Y, R.R.Ramos y. 2007.** *Modelo de Evaluación del Proceso de Desarrollo del Software Educativo*. La Habana : s.n., 2007. Investigativo. Universidad de las Ciencias Informáticas.
- ✚ **Piattini, M. 1996.** Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Madrid : Rama, 1996.
- ✚ **Pressman, Roger S. 2002.** *Ingeniería de Software. Un enfoque práctico*. 5ta edición. New York, Mac Graw Hill : s.n., 2002.
- ✚ **Ricardo, Febe Angel Ciudad. 2007.** *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI*. La Habana : s.n., 2007. pág. 31.
- ✚ **Rodríguez, L.D.M. Alvarez y L.K.C. 2006.** 2006.
- ✚ **Ruz, Fidel Castro. 2004.** *Ciencia, Tecnología y Sociedad. Hacia un desarrollo sostenible en la Era de la Globalización*. La Habana : Científico-Técnica, 2004.



- ✚ **Sánchez, Jaime. 1999.** *Construyendo y aprendiendo con el computador.* Santiago de Chile : Proyectos Enlaces-MECE, 1999. Centro Zonal Universidad de Chile.
- ✚ **Squires, McDougall. 1997,2007.** *Cómo elegir y utilizar software educativo.* Madrid : Ediciones Morata, 1997,2007.
- ✚ **Valenzuela, Teresa. 2008.** *Crecimiento vertiginoso del software educativo en Cuba.* La Habana, 17 de Octubre de 2008. www.radiorebelde.com.
- ✚ **Zambrano, J. 1995.** *Enseñanza asistida por computador y producción de software educativo(PROSODS).* Caracas, Venezuela : Imprenta Universitaria, 1995. Universidad Central de Venezuela.



Anexos

Anexo #1: Documentos de salida de los procesos del ciclo de vida

Tabla 11: Documentos de salida de los procesos del ciclo de vida. (Cataldi, 2000)

Procesos	Documentos de salida
Proceso de identificación de la necesidad educativa.	<ul style="list-style-type: none"> Definición del marco educativo y comunicacional.
Proceso de selección del modelo de ciclo de vida.	<ul style="list-style-type: none"> Ciclo de vida adoptado.
Proceso de iniciación, planificación y estimación del proyecto.	<ul style="list-style-type: none"> Plan de gestión del proyecto.
Proceso de seguimiento y control del proyecto (programa).	<ul style="list-style-type: none"> Análisis de riesgos y plan de contingencias. Registro histórico del proyecto.
Proceso de gestión de calidad del software.	<ul style="list-style-type: none"> Plan de garantía de calidad. Recomendaciones de mejora de calidad.
Proceso de exploración de conceptos.	<ul style="list-style-type: none"> Informe de necesidades. Posibles soluciones factibles.
Proceso de asignación del programa (sistema).	<ul style="list-style-type: none"> Especificación de requisitos funcionales de hardware y software. Especificación de interfaces del sistema o programa. Descripción funcional. Arquitectura.
Proceso de análisis de requisitos educativos	<ul style="list-style-type: none"> Especificación de los objetivos y estructuración de conceptos. Selección de contenidos y pertenencia.



Proceso de análisis de requisitos del software		<ul style="list-style-type: none"> • Especificación de requisitos del software, de interfaces de usuario y de otros software. Interfaces del hardware y con el sistema físico.
Proceso de diseño	De los contenidos	<ul style="list-style-type: none"> • Identificación de los procesos mentales a estimular. • Definición de las actividades a realizar por los alumnos. • Jerarquización de los conceptos.
	Del Software	<ul style="list-style-type: none"> • Descripción del diseño del software y de la arquitectura. • Descripción del flujo de información, bases, interfaces y algoritmos.
Proceso de implementación e integración de módulos		<ul style="list-style-type: none"> • Datos para las pruebas. • Documentación para el sistema o programa y del usuario. • Plan de integración.
Proceso de instalación		<ul style="list-style-type: none"> • Plan de instalación. • Informe de instalación.
Proceso de operación y soporte		<ul style="list-style-type: none"> • Histórico de pedidos y soporte
Proceso de mantenimiento		<ul style="list-style-type: none"> • Recomendaciones de mantenimiento
Proceso de retiro		<ul style="list-style-type: none"> • Plan de retiro
Proceso de verificación y validación		<ul style="list-style-type: none"> • Plan de verificación y validación. • Plan de pruebas. • Especificación y resumen de la prueba. • Software aprobado.



Proceso de evaluación de los prototipos del software	<ul style="list-style-type: none"> • Diseño del instrumento de evaluación. • Resumen de la prueba. • Selección de la muestra.
Proceso de evaluación interna y externa del software	<ul style="list-style-type: none"> • Diseño del instrumento de evaluación. • Resumen de la prueba. • Selección de la muestra.
Proceso de evaluación contextualizada	<ul style="list-style-type: none"> • Diseño de la experiencia. • Definición de los grupos de control y experimental.
Proceso de Configuración	<ul style="list-style-type: none"> • Plan de gestión de la configuración
Proceso de documentación técnica	<ul style="list-style-type: none"> • Plan de documentación técnica
Proceso de documentación didáctica	<ul style="list-style-type: none"> • Plan de documentación didáctica
Proceso de formación y captación del personal	<ul style="list-style-type: none"> • Plan de formación y captación



Anexo #2: Etapas del Ciclo de Vida

Tabla 12: Etapas del ciclo de vida

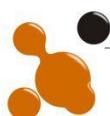
Etapas	Descripción
Factibilidad (FAC)	En esta etapa se define el producto de software y se determina su factibilidad en el ciclo de vida desde la perspectiva de la relación costo–beneficio, así como las ventajas y desventajas respecto a otros productos.
Requisitos del sistema (RES)	En esta etapa se definen las funcionalidades requeridas para el desarrollo del sistema (o programa), las interfaces y el tipo de diseño.
Especificación de requisitos del prototipo (REP)	Consiste en especificar las funciones requeridas, las interfaces y el rendimiento para el prototipo. Aquí se consideran incrementos en porcentajes de la funcionalidad total del sistema.
Diseño del prototipo (DPR)	Es poner en ejecución el plan del prototipo, ya que una vez fijadas las restricciones con el usuario, hay que mostrar el mismo funcionando, aunque sean sólo algunas funcionalidades restringidas. Aquí, hay que hacer un análisis de cómo se va a trabajar, qué módulos se van a hacer, con qué lógica y qué funciones se van a usar.
Diseño detallado del prototipo (DDP)	Esta etapa es una especificación verificada de la estructura de control, la estructura de los datos, las relaciones de interfaces, el tamaño, los algoritmos básicos y las suposiciones de cada componente del



	programa. En esta etapa no sólo se definen y sino que se documentan los algoritmos que llevarán a cabo la función a realizar por cada uno de los módulos.
Desarrollo del prototipo (codificación) (DEP)	Consiste en realizar la codificación o diseño detallado, en forma legible para la máquina.
Implementación y prueba del prototipo (IPP)	Consiste en lograr un funcionamiento adecuado del producto software en el sistema informático, funcionando operacionalmente, incluyendo objetivos tales como la conversión del programa y datos (si la hubiere), la instalación y el entrenamiento. La prueba debe asegurar que se han probado todas las sentencias del mismo, y que en las funciones externas se han realizado pruebas que aseguren que la entrada definida produce los resultados que se esperan realmente.
Refinamiento iterativo de las especificaciones del prototipo (RIT)	Es un aumento de la funcionalidad del sistema, para luego volver REP a fin de aumentar la funcionalidad del prototipo o continuar.
Diseño del sistema final (DSF)	Consiste en ajustar las restricciones o condiciones finales e integrar los últimos módulos.
Implementación del sistema final (ISF)	Es el sistema informático funcionando operativamente, incluyendo objetivos como conversión del programa y datos, (si la hubiere), la instalación y la capacitación del personal.



Operación y mantenimiento (OPM)	Es la puesta en funcionamiento del sistema informático, objetivo que se repite para cada actualización.
Retiro (RET)	Es una transición de las funciones realizadas para el producto y sus sucesores. Luego se definen los procesos básicos para este ciclo de vida, y las actividades para cada uno de ellos. Los procesos incluyen aquellos concernientes al desarrollo de software y los específicos teniendo en cuenta los aspectos educativos, aunque en la propuesta, no se define una teoría educativa en particular, sino que las actividades permiten ver un enfoque cognitivista-constructivista.



Anexo #3: Planilla Modelo Pedagógico

Modelo Pedagógico

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos

1. Introducción

<< El propósito de este documento es mostrar el contenido de enseñanza, así como características para la buena práctica del docente. >>

1.1. Propósito

<<Especifica el objetivo del documento. >>

1.2. Alcance

<<Una descripción breve del alcance del documento. >>

1.3. Definiciones, acrónimos y abreviaturas

<<Esta subsección describe las definiciones de todos los términos, acrónimos y abreviaturas necesarias para interpretar adecuadamente el documento. >>



1.4. Referencias

<<Esta subsección proporciona la lista completa de todos los documentos referenciados en el documento de Visión. Identifica cada documento por el título, número de informe, fecha y organización que lo desarrolla. >>

1.5. Visión General

<<Descripción del contenido del documento y organización. >>

2. Problema instruccional detectado

3. Listado de necesidades

4. Estudio contextual

<< Esta sección se centrará en la manera en la que el profesorado realiza sus intervenciones educativas, considerando todas las circunstancias que inciden en su actuación. Para ello el profesor dará un marco general del programa educativo. >>

5. Caracterización de la Audiencia

5.1. Generales

5.2. Específicas

5.3. Estilos de aprendizajes

5.4. Otras informaciones

6. Tareas Instruccionales

6.1. Desglose de las tareas instruccionales

7. Objetivos instruccionales

<<En esta sección se indicará lo que uno desea enseñar y espera que los alumnos aprendan en una determinada clase. Está basado en las destrezas, los conceptos y los valores que se quieren desarrollar en los estudiantes. >>



7.1. Clasificados

8. Diseño de Secuencia de Contenidos

<< En esta sección cada secuencia de contenidos estará diseñada en función de unos objetivos a lograr por el alumno, la selección de unos contenidos (conceptuales, procedimentales y actitudinales), el desarrollo de un conjunto de actividades, diseñadas en función de una propuesta metodológica concreta, y uno o varios instrumentos de evaluación. >>

8.1. Secuenciados por niveles de prioridad y complejidad

<< En esta subsección los contenidos estarán separados por niveles de prioridad y complejidad >>

9. Diseño de los contenidos (mensaje)

<<Esta sección, permitirá lograr una correcta elaboración del diseño del contenido, para ello se debe tener en cuenta: el ámbito disciplinario, que corresponde al área de conocimiento de la materia, el ámbito metodológico, entendido como forma de facilitar el aprendizaje y el ámbito tecnológico, que establece las tecnologías para la elaboración del material. Seguidamente se seguirá el siguiente programa:

- *Detección de necesidades (DIAGNÓSTICO)*
- *Determinación de objetivos.*
- *Diseño de actividades y estrategias.*
- *Organización y temporalización.*
- *Evaluación. >>*

10. Desarrollo de la estrategia instruccional

<<Esta sección desarrollará la forma de organizar secuencialmente y llevar a cabo las situaciones de enseñanza-aprendizaje. Además se analizarán los medios que deben utilizarse y la forma en la cual deben agruparse los estudiantes para la instrucción. >>



10.1. Mapa conceptual

<<En esta subsección se mostrará un mapa conceptual, que no es más que un instrumento de representación del conocimiento sencillo y práctico, que permite transmitir con claridad mensajes conceptuales complejos y facilita tanto el aprendizaje como la enseñanza. Esta técnica permitirá la organización y representación gráfica del conocimiento. >>



Anexo #4: Planilla Documento Visión

Documento Visión

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos

1. Introducción

<< El propósito de este documento es recopilar, analizar, y definir las necesidades de alto nivel y características del sistema necesarias para los implicados y usuarios finales. Los detalles de cómo el sistema satisface estas necesidades se detallan en los casos de uso y en la especificación suplementaria. >>

1.1. Propósito

<< Especifica el objetivo del documento. >>

1.2. Alcance

<< Una descripción breve del alcance del documento. >>



1.3. Definiciones, acrónimos y abreviaturas

<< Esta subsección describe las definiciones de todos los términos, acrónimos y abreviaturas necesarias para interpretar adecuadamente el documento de Visión. >>

1.4. Referencias

<< Esta subsección proporciona la lista completa de todos los documentos referenciados en el documento de Visión. Identifica cada documento por el título, número de informe, fecha y organización que lo desarrolla. >>

1.5. Visión General

<< Descripción del contenido del documento y organización. >>

2. Análisis didáctico

<<Se realizarán algunas críticas del contenido, estructura y forma del programa educativo. Además quedará plasmado con exactitud el por qué de la existencia del proyecto educativo; para qué se enseña; qué se enseña; cómo debe desarrollar el proceso para lograr el aprendizaje consciente del estudiante; cómo comprobar el cumplimiento de los objetivos; cómo lograr la interrelación de los contenidos. >>

3. Información sobre los destinatarios del audiovisual.

<<Se dará una breve descripción sobre los destinatarios que usarán la aplicación educativa. >>

4. Definición y redacción de los objetivos educativos.

<<Establecer las finalidades del producto (para qué, con qué finalidad, qué queremos conseguir,...) En conclusión qué resultados se quieren alcanzar con el uso del material audiovisual. >>

5. Orientaciones educativas: uso del producto

<<Sugerencias de actividades educativas previas y posteriores al uso del producto. >>

<<Sugerencias para la evaluación del uso del producto. >>



6. Diseño del material

6.1. Tema a tratar

<<Determinar con claridad el contenido sobre el que va a tratar el audiovisual. Es preferible elegir un tema específico (ej. el abuso en el consumo de agua) antes que un tema general (ej. la educación ambiental). >>

6.2. Destinatarios

<<Es muy importante definir los destinatarios del material que vamos a elaborar de un modo genérico pero preciso (edad, curso, necesidades educativas, etc.) >>

7. Información del producto

7.1. Oportunidad del negocio

<< Se describe de forma breve la oportunidad de negocio que proporciona este proyecto >>

7.2. Informe del problema

<< Informe resumido del problema que se pretende resolver. Se puede usar el siguiente formato:

El problema de	<i>Describir el problema</i>
Afecta	<i>Los implicados afectado por el problema</i>
El efecto del problema es	<i>¿Cuál es el impacto del problema?</i>
Una solución con éxito debería ser	<i>Lista de algunos beneficios claves de una solución con éxito</i>

7.3. Informe del producto

<< Informe resumido, al nivel más alto, del sistema a desarrollar dentro del ámbito de negocio. Se puede usar el siguiente formato:



<i>Para</i>	<i>Cliente final</i>
<i>Quién</i>	<i>Informe de la necesidad u oportunidad</i>
<i>El (nombre del producto)</i>	<i>Es un [categoría del producto]</i>
<i>Qué</i>	<i>Informe del beneficio principal, esto es, la razón irresistible para comprar</i>
<i>Distinto de</i>	<i>Principal alternativa de la competencia</i>
<i>Nuestro producto</i>	<i>Informe de las principales diferencias</i>

7.3.1. Recursos necesarios, medios y tiempo de interacción

<<En esta sección quedarán especificados los recursos necesarios, medios indispensables y el tiempo que durará la interacción con el software educativo>>

8. Descripción de los implicados y usuarios finales

<< Para desarrollar sistemas de software efectivos que reúnan las necesidades de los implicados y usuarios reales, es necesario identificar e involucrar a todos como parte del proceso de Modelado de Requisitos. Esta sección muestra un perfil de los implicados y usuarios relacionados con el proyecto, y describe sus problemas principales. Debe justificar por qué los requisitos de los implicados y usuarios son necesarios>>

8.1. Entorno de los usuarios

<< Describe los usuarios estimando su tamaño y estudiando el crecimiento de usuarios potenciales. Además, para cada usuario hay que determinar quién es, su nivel educativo, nivel de uso de aplicaciones informáticas, experiencia, plataformas usadas, plataformas que va a usar en el futuro, expectativas de usabilidad, aplicaciones con las que necesita interfaz, documentación necesaria. >>



8.1.1. Resumen de los involucrados

<< Se describen los implicados que hay que considerar en el proceso de desarrollo y que no son usuarios finales. >>

<i>Nombre</i>	<i>Descripción</i>	<i>Responsabilidad</i>
<i>Nombre del tipo de implicado</i>	<i>Breve descripción</i>	<i>Los intereses del implicado en el proyecto</i>

8.1.2. Perfiles de los involucrados

<< Se describe cada implicado en el sistema cumplimentando la siguiente tabla. >>

<i>Representante</i>	<i>¿Quién es el implicado representativo para el proyecto?</i>
<i>Descripción</i>	<i>Breve descripción del tipo de implicado</i>
<i>Tipo</i>	<i>Calificar los conocimientos técnicos, y los grados sofisticados (gurú, experto, usuario casual, etc.)</i>
<i>Responsabilidades</i>	<i>Intereses del implicado</i>
<i>Criterios de éxito</i>	<i>¿Cómo el implicado define el éxito?</i>
<i>Implicación</i>	<i>¿Cómo está involucrado en el proyecto?</i>
<i>Comentarios/Cuestiones</i>	<i>Problemas que interfieren con el éxito del proyecto y cualquier otra información relevante.</i>

8.1.3. Necesidades principales de los involucrados y usuarios

<<Lista de los problemas principales con las soluciones existentes según los implicados.

Clarificar los siguientes conceptos para cada problema:

- *¿Cuáles son las razones para el problema?*
- *¿Cómo se resuelve actualmente?*



- ¿Qué soluciones quieren los implicados o usuarios?

Es importante entender la relativa importancia de los implicados y usuarios en la resolución de cada problema. >>

<i>Necesidad</i>	<i>Prioridad</i>	<i>Problema</i>	<i>Solución actual</i>	<i>Solución propuesta</i>

<<Se pasa un cuestionario a los usuarios para que expresen sus necesidades e identifiquen los problemas que se encuentran. El cuestionario se anexa al Documento de Visión. >>

9. Visión general del producto

<< Esta sección proporciona una visión de alto nivel de las capacidades del producto, interfaces y otras aplicaciones, y configuraciones de sistemas. Esta sección estará dividida en cinco subsecciones:>>

9.1. Perspectivas del producto

<<Esta subsección puede poner la perspectiva de este producto relativo a otros productos o ambientes de usuarios. Si el producto es independiente y totalmente autocontrolado declárelo aquí. Si el producto es un componente de un gran sistema, entonces esta subsección puede relatar cómo estos sistemas interactúan y puede identificar interfaces relevantes entre los sistemas. Una forma fácil para mostrar los principales componentes del gran sistema, interconexiones e interfaces externas, son los diagramas de bloque. >>

9.2. Funciones del producto

<<Listar las funciones del producto para que el cliente o cualquier persona tenga conocimiento de lo que hará el producto audiovisual. >>

9.3. Suposiciones y dependencias

<<Lista cada uno de los factores que afectan a las características indicadas en el Documento Visión. Lista las suposiciones, que si están cambiadas, alterarán el Documento Visión. Por ejemplo, una suposición puede indicar que un sistema operativo específico



estará disponible para el hardware señalado para el producto del software. Si el sistema operativo no está disponible, el Documento Visión necesitará cambiar. >>

9.4. Coste y precio

<<Para comercializar el producto a clientes externos y para muchas aplicaciones en casa, los asuntos de costo y precio pueden directamente impactar la definición e implementación del producto. En esta sección, se registran las restricciones de costo y precio que son relevantes. Por ejemplo: costo de distribución (# de discos, #-de CD, etc.) u otros costos para restricciones de buena comercialización (manuales) otros elementos de costo (manuales, ensamblaje packaging.) >>

9.5. Licencia e instalación

<< Los conceptos de licencia e instalación pueden también afectar directamente a los esfuerzos de desarrollo. Los requisitos de instalación pueden también afectar codificando o creando las necesidades para instalaciones de software separadas. >>

10. Resumen de capacidades

<<Resumen de los principales beneficios y características que el producto debe tener. Por ejemplo, para un sistema de soporte al cliente se puede usar esta parte para enunciar los problemas de documentación, camino y estado reportando con el nivel de detalle que cada función requiere.

Organizar las funciones así como la lista es entendible para el cliente o cualquier otra persona que lea el documento por primera vez. Una simple tabla lista las claves para el beneficio y soporta las características. >>

11. Características básicas del producto

<<Lista y breve descripción de las características básicas del producto. Estas características básicas son las capacidades elementales con las que debe cumplir el producto para satisfacer al cliente. Cuando se desarrolle el modelo de Casos de uso se describen el resto de las especificaciones necesarias.



Este documento puede ser revisado por cualquier involucrado por lo tanto el lenguaje a utilizar no debe ser técnico pero debe contener todos los detalles que pueda necesitar el que desarrolle el modelo de casos de uso. Se recomienda no tener menos de 25 ni más de 99 características. >>

12. Requisitos para la documentación

<<Esta sección describe la documentación que se debe desarrollar para conseguir el éxito de la aplicación. >>

12.1. Manual de usuario

<<Describe el propósito y contenidos del Manual de Usuario. Discute la longitud, nivel de detalle, necesidades de índice, glosario de términos, tutoriales contra manual de usuario, etc. Restricciones de formato e impresión. >>

12.2. Ayuda online

<<Muchas aplicaciones proporcionan un sistema de ayuda online para asistir al usuario. >>

12.2.1. Guías de instalación, configuración y fichero Readme

<<Un documento que incluye instrucciones de instalación y guías de configuración es importante para ofrecer una solución completa. También se incluye como un componente estándar un fichero ReadMe. >>

12.2.2. Etiquetado y empaquetado

<<Esta sección define las necesidades y tipos de etiquetados que se deben incorporar en el código. Ejemplos incluyen copyright y patentes, logos corporativos, iconos...>>



Anexo #5: Planilla Especificación de Requisitos

Especificación de Requisitos

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos

1. Introducción

<< El propósito de este documento es reunir requerimientos, restricciones e información que resulta difícil reflejar en los Casos de Uso o en el Glosario. >>

1.1 Propósito

<< Especifica el objetivo del documento. >>

1.2 Alcance

<< Proyectos con los que se involucra la Especificación. >>

1.3 Definiciones, Acrónimos y Abreviaturas

<<Esta subsección describe las definiciones de todos los términos, acrónimos y abreviaturas necesarias para interpretar adecuadamente el documento de Especificación de Requisitos>>



1.4 Referencias

<< Lista de documentos a los que se hace referencia en la Especificación >>

2. Requisitos Funcionales

<<En esta sección se enumerarán todos los requisitos funcionales que presente el sistema educativo. Deben relacionarse cada uno de los requerimientos funcionales y sus respectivas descripciones. Pueden ser agrupados por la clasificación que se estime conveniente. Dentro de las categorías y agrupaciones definidas los requerimientos deberán especificarse como sigue:

Nombre del Requerimiento # 1. Descripción del requerimiento;

Nombre del Requerimiento # 2. Descripción del requerimiento; y así sucesivamente.

A continuación se mostrarán algunos de los requerimientos funcionales que no deben faltar durante el análisis de cualquier software educativo: >>

2.1. Eficacia didáctica

<<Esta subsección facilitará el logro de los objetivos. Un material formativo ante todo debe resultar eficaz, debe facilitar el logro de los objetivos instructivos que pretende: localizar información, obtener materiales, archivarlos e imprimirlos, encontrar enlaces, consultar materiales didácticos, realizar aprendizajes...>>

2.2. Flexibilización del aprendizaje

<<Incluye diversos niveles, itinerarios... Los materiales didácticos se adaptarán a las características específicas de los estudiantes (diferencias en estilos de aprendizaje, capacidades...) y a los progresos que vayan realizando los usuarios, para que hagan un máximo uso de su potencial cognitivo. Esta adaptación se manifestará especialmente en la forma de tutorar, en la progresión de las actividades que se presenten a los estudiantes y en la profundidad de los contenidos que se trabajen. >>



2.3. Calidad y estructuración de los contenidos

<<Al margen de otras consideraciones psicológicas y pedagógicas sobre la selección y estructuración de los contenidos según las características de los usuarios, hay que tener en cuenta las siguientes cuestiones:

- La información que se presenta es correcta en extensión, rigor científico y actualidad, se presenta bien estructurada diferenciando adecuadamente: datos objetivos, opiniones y elementos fantásticos.
- Los textos no tienen faltas de ortografía y la construcción de las frases es correcta.
- La información, organizada en bloques y páginas, se presenta mediante párrafos breves para facilitar la lectura.
- Proporciona enlaces entre distintos conceptos, y están actualizados. >>

2.4. Navegación

<<Mapa de navegación lógico y estructurado; metáforas intuitivas, atractivas y adecuadas a los usuarios. El entorno debe ser transparente, permitiendo al usuario saber siempre donde está y tener el control de la navegación. >>

3. Requisitos No Funcionales

<<En esta sección se enumerarán todos los requisitos no funcionales que presente el SWE.

Deben enumerarse cada uno de los requerimientos no funcionales y sus respectivas descripciones. Podrán clasificarse o agruparse según convenga; pero deberán especificarse como mínimo los aspectos siguientes:

- **Performance:** esquematiza las características de ejecución del sistema [Tiempo de respuesta para una transacción (promedio y máximo), Rendimiento (Ejemplo: transacciones por segundo), Capacidad (Ejemplo: el número de clientes o transacciones que el sistema puede hospedar), Utilización de recursos: memoria, disco, comunicación, etc.]
- **Restricciones de diseño:** esta sección indica cualquier restricción de diseño sobre el sistema a



ser construido. Las restricciones de diseño representan decisiones de diseño que han sido acordadas y deben ser adheridas. Ejemplos: incluye lenguajes de programas, requerimientos del proceso de programas, uso reglamentado de herramientas de desarrollo, componentes adquiridos, etc.

- **Requerimientos de documentación online de usuarios y ayudas del sistema:** describe los requerimientos de ayuda y documentación online.
- **Requerimientos Técnicos de los Recursos Audiovisuales**
- **Características.** Se definen teniendo en cuenta los formatos y propiedades de cada uno de los tipos de recursos audiovisuales. Para todos los casos se especifica el formato y la limitante de tamaño de los ficheros (KB, MB). La especificación de propiedades varía según el recurso.

Tipo de Recurso	Propiedades
Vídeo	Resolución (Dimensiones)
Audio (locuciones, efectos de sonido, música)	Frecuencia, Canal
Imagen	Resolución (Dimensiones)

Dentro de las categorías y agrupaciones definidas los requerimientos no Funcionales deberán especificarse como sigue:

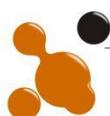
Nombre del Requerimiento no Funcional # 1. Descripción del requerimiento

Nombre del Requerimiento no Funcional # 2. Descripción del requerimiento; y así sucesivamente.

A continuación se mostrarán algunos de los requerimientos NO funcionales que debe cumplir cualquier software educativo: >>

3.1. Facilidad de uso del entorno

<<Los materiales deben resultar agradables, fáciles de usar y autoexplicativos, de manera que los usuarios puedan utilizarlos inmediatamente, y descubran su dinámica y sus posibilidades, sin tener que realizar una exhaustiva lectura de los manuales ni largas tareas previas de configuración. El usuario debería conocer en todo momento el lugar del programa donde se encuentra y las opciones a su alcance, y debería poder moverse en él según sus preferencias. Un "sistema de ayuda", accesible desde el mismo material, debería solucionar las dudas>>



3.2. Entorno Audiovisual

<<Presentación, estructura de las pantallas, composición, tipografía, colores, disposición de los elementos del producto, estética...

- Presentación atractiva y correcta. Indicará también la resolución óptima para su visualización (800x600...)

- Diseño claro y atractivo de las pantallas, sin exceso de texto, destacando lo importante.

- Calidad técnica y estética en sus elementos: títulos, barras de estado, frames, menús, barras de navegación, ventanas, iconos, botones, textos, hipertextos, formularios, fondos... Algunas de estas características son en caso de multimedia educativa. >>

3.3. Facilidad de acceso e instalación de programas y complementos.

<<La instalación y desinstalación del software debe ser sencilla, rápida y transparente. En el material quedará orientada la instalación de los drivers y visualizadores necesarios. >>

3.4. Funcionamiento del entorno

<<Fiabilidad, velocidad adecuada, seguridad... El material debe visualizarse bien en los distintos navegadores, presentar una adecuada velocidad de respuesta a las acciones de los usuarios al mostrar informaciones, videos, animaciones... Si se trata de un programa informático detectará la ausencia de periféricos necesarios y su funcionamiento será estable en todo momento. >>

3.5. Restricciones en el diseño y la implementación

<<Esta sección debe indicar cualquier restricción de diseño en el sistema a construir. Las restricciones representan decisiones de diseño que se han tomado y a las cuales es necesario adherirse. (Ej. lenguajes de programación, requisitos de proceso de software, el uso prescrito de herramientas de desarrollo, restricciones de arquitectura y diseño, componentes comprados, las bibliotecas de la clase, etc. >>



3.6. Requerimientos de apariencia o interfaz externa

<<Esta sección se definen las interfaces que deben soportadas por la aplicación de software educativo. Debe contener la especificidad adecuada, protocolos, puertos y direcciones lógicas, etc., para que dicho software pueda desarrollarse y verificarse contra los requisitos de la interfaz. >>



Anexo #6: Planilla Glosario de Términos

Glosario de Términos

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1 Propósito

<<Especifica el propósito del glosario>>

1.2 Alcance

<<Breve descripción del alcance del glosario; proyectos con los que se relaciona y cualquier otro detalle que pueda facilitar su comprensión. >>

1.3 Referencias

<<Lista completa de todos los documentos referenciados en el Glosario. >>

1.4 Resumen

<<Describe que contiene el Glosario. >>



2. Definiciones

<<Los términos definidos a continuación son la parte principal de este documento. Estos son definidos en cualquier orden pero generalmente un orden alfabético provee un mejor acceso. >>

2.1 <aTerm>

<< La definición de <aTerm>se especifica aquí. Se debe especificar todos los detalles necesarios para comprender el término. >>

2.2 <otroTerm>

<<La definición de <otroTerm>se especifica aquí. Se debe especificar todos los detalles necesarios para comprender el término. >>

2.3 <Grupo deTerms>

<<Algunos conceptos se definen a partir del concepto de varios términos. En esta sección se agrupan bajo un término específico la definición de todos los que lo definen a él. >>

2.3.1 <GroupoTerm>

<<La definición de <GroupoTerm>se especifica aquí. Se debe especificar todos los detalles necesarios para comprender el término. >>

2.3.2 <otroGroupoTerm>

<<La definición de <otroGroupoTerm>se especifica aquí. Se debe especificar todos los detalles necesarios para comprender el término. >>



Anexo #7: Planilla Descripción para la Arquitectura de Software

Descripción para la Arquitectura de Software

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de contenidos:

1 Introducción

1.1 Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar qué se ve afectado por este documento. >>

1.2 Definiciones, Acrónimos y Abreviaturas

<<Especifique las definiciones, abreviaciones y siglas que tienen que ver con este documento a fin de su correcto entendimiento. >>

2 Resumen Arquitectónico

2.1 Estilo Arquitectónico

*<<En este apartado se debe responder la siguiente pregunta:
¿Qué estilo de arquitectura de software está siendo usado? >>*



2.2 Objetivos de la Arquitectura

<<En esta sección se deben detallar objetivos del software que tienen relación con la arquitectura: expansibilidad, portabilidad, funcionalidad, distribución, reutilización, etc. >>

3 Componentes Significativos de la Arquitectura del Sistema

<<Los componentes de este sistema deben estar definidos claramente en los diagramas de componentes hechos con UML.

Describa brevemente cada componente del sistema que sea relevante para la arquitectura del sistema. Enfóquese en los detalles arquitectónicos tales como mecanismos de comunicación, aspectos del entorno que afecten el desarrollo, y concurrencia. >>

4 Vista de Casos de Uso

<<Si constituyen alguna funcionalidad significativa, se deben listar los Casos de Uso o escenarios del Modelo de Casos de Uso; si poseen características arquitectónicas respectivas, como un gran número de elementos o si ilustran algún punto crítico de la arquitectura. Si los casos de uso están empaquetados deben listarse por paquetes. >>

5 Vista Lógica

<<Se detallan las partes del modelo de diseño que son significativas arquitectónicamente, como son su descomposición dentro de subsistemas y paquetes de servicios; y debe presentarse para cada paquete, su descomposición dentro de clases y ventajas de las clases.

Introducir las clases significativas arquitectónicamente y describir sus responsabilidades, así como sus relaciones, funciones y características. >>

5.1 Repartición del Procesamiento

<<Se debe describir la descomposición del modelo de diseño en términos de su rango de paquetes y capas. En este apartado incluir algunos diagramas reflejando los paquetes de nivel alto, así como su dependencia y sus capas. >>

5.2 Paquetes de Diseño significativos Arquitectónicamente

<<Esta sección debe contener una subsección con el nombre del paquete, una breve descripción



y un diagrama con todas las clases y paquetes significativos, contenidos dentro del paquete.

Para cada clase significativa en el paquete, refleje su nombre, una breve descripción y, opcionalmente, alguna descripción de sus principales responsabilidades, funciones y características. >>

5.3 Realización de los Casos de Uso

<<Se debe ilustrar cómo normalmente el software opera, presentando algunos casos de uso escogidos, y expone cómo los distintos elementos del modelo de diseño sobrellevan a su funcionalidad. >>

6 Vista de Procesos

<<Se describe la descomposición del sistema dentro de procesos y grupos de procesos. Se debe organizar el apartado por grupos de procesos donde se reflejen sus comunicaciones o interacciones. En primer lugar se debe describir los principales modos de comunicación entre procesos, como: pases de mensajes, etc. Para cada red de procesos se debe incluir una subsección con la siguiente información:

Su nombre.

Los procesos involucrados.

Las interacciones entre los procesos en la forma de diagramas de colaboración, en los cuales los objetos son los procesos reales que completan su propio vínculo de control. Para cada proceso, describa brevemente su comportamiento, su ciclo de vida y sus características de comunicación. >>

7 Vista de Implementación

<<En esta vista se debe mostrar en general las dependencias y cómo se implementan los componentes físicos del sistema, agrupándolos en subsistemas organizados en capas y jerarquías. >>



7.1 Visión General

<<Nombrar y definir los contenidos de las distintas capas, las reglas que controlan la inserción dentro de una capa y las restricciones entre capas. Además, se debe reflejar un diagrama de componentes que muestre las relaciones que existen entre capas. >>

7.2 Capas

*<<En esta sección para cada capa se debe incluir una subsección con la siguiente información:
Su nombre.*

Una lista de los subsistemas ubicados en la capa, con nombre, abreviación y una breve descripción.

Un diagrama de componentes que muestre los subsistemas y sus dependencias importantes.

Su relación con elementos de la Vista Lógica. >>

8 Vista de Implantación

<<Describir una o más configuraciones de redes físicas (hardware) sobre las cuales el software será desarrollado. Se debe indicar los nodos físicos para cada configuración (computadores, CPU) que ejecutan el software, y sus interconexiones (bus, LAN, punto a punto, etc.). También se incluye un mapa de los procesos de la vista de procesos, dentro de los nodos físicos. Para cada configuración física de red incluya una subsección con la siguiente información:

- Su nombre.*
- Un diagrama de entrega que ilustre la configuración.*
- Mapa de procesos para cada procesador. >>*

9 Vista de Datos

<<Reflejar la perspectiva del almacenamiento de datos constantes en el sistema. Esta sección es opcional. Se debe describir una apreciación global del modelo de los datos y su organización por lo que se refiere a las tablas, vistas y almacenamiento de los procedimientos que proporcionan la persistencia al sistema. >>



10 Integración

10.1. Integración de los Componentes y su Comunicación

<<En este apartado se debe responder la siguiente pregunta:

¿Cómo serán integrados los componentes? Específicamente, ¿cómo se comunicarán? >>

10.2. Mecanismos de la Arquitectura para Futuras Modificaciones o Extensiones

<<En este apartado se debe responder la siguiente pregunta:

¿Qué mecanismos arquitecturales se utilizan para facilitar futuras extensiones o modificaciones?>>

11 Lista de Control de la Arquitectura

<<Evalúe su arquitectura respecto a cada uno de sus objetivos:

Facilidad de Integración

Expansibilidad >>

12 Aseguramiento de la Calidad

12.1 Alcance del Plan de Calidad

<<Se debe identificar los componentes y aspectos del sistema que deberán ser evaluados para asegurar que los objetivos de calidad se han alcanzado. >>



Anexo #8: Planilla Mapa de Navegación

Mapa de Navegación

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos

1. Introducción

1.1 Propósito

<<Especifica el propósito del Mapa de Navegación >>

1.2 Alcance

<<Breve descripción del alcance del Mapa de Navegación; proyectos con los que se relaciona y cualquier otro detalle que pueda facilitar su comprensión. >>

1.3 Referencias

<<Lista completa de todos los documentos referenciados en el Mapa. >>

1.4 Resumen

<<Describe qué contiene el Mapa. >>



2. Sistema de Navegación

<<El sistema de navegación incluye unos aspectos algorítmicos y estructurales que reflejan una primera aproximación a la estructura del programa, y se concretarán en diversos gráficos y diagramas:

2.1. Diagrama general del programa (mapa de navegación)

<<Reproduce la estructura básica de su algoritmo. Se acostumbra a representar en forma de diagrama de flujo, y debe de ir acompañado de una breve descripción de los módulos globales que lo integran:

- Módulos de presentación y de gestión de menús:

<<Comprenden las pantallas de presentación y despedida del programa y las pantallas de gestión de los menús principales. >>

- Módulos de ayuda:

<<Gestionan las ayudas a los alumnos. Hay que determinar las formas de acceso a estas ayudas, que pueden ser:

- *Ayudas sobre el funcionamiento del programa.*
- *Ayudas didácticas, sobre los contenidos. >>*

- Módulos auxiliares

<<Por ejemplo: gestión de posibles modificaciones de parámetros, utilidades para los alumnos (calculadora, diccionario...), etc. >>

2.2. Organización de los menús

<<Tras determinar si los menús estarán organizados según un entorno tradicional o según un entorno Windows y en forma de menús desplegables (top down), se diseñará el árbol de las opciones que el programa ofrecerá a los usuarios. >>



2.3. Parámetros de configuración del programa.

<<La posibilidad que los profesores y los alumnos puedan adaptar algunos aspectos del programa a sus circunstancias concretas es una característica cada vez más valorada en los programas. >>

2.4. Itinerarios pedagógicos

<< (Implícitos del programa, explícitos del alumno). Representa la secuencia en que se presentarán las actividades y sus posibles bifurcaciones en función de los comportamientos (acciones, errores, etc.) de los usuarios. Se procurará dejar el máximo control posible al alumno. >>

2. Entorno Audiovisual

<<Se realizará el diálogo entre los estudiantes y el programa

- ¿Qué metáfora se utilizará para la navegación?

- ¿Cómo se presentará la información en las páginas? Composición de los elementos (textuales, multimedia...): textos, tablas, hipervínculos, fondos, tipografía, colores, íconos, botones, barras de herramientas, marcos...

- ¿Es coherente el entorno? Facilita el acceso rápido a la información y sus elementos realizan siempre la misma función. >>

3. Actividades e Interacciones

<<Además de la interactividad propia de la navegación por la aplicación educativa se dará una breve descripción de cada una de las actividades que se desarrollarán en la aplicación para darle un mejor entendimiento al usuario final. Estas actividades facilitan el intercambio de informaciones entre los alumnos y la máquina que permite que las acciones de los estudiantes puedan ser valoradas y tratadas por el programa. Además podrán existir algunas actividades interactivas, como por ejemplo: Bidireccionalidad comunicativa. ¿Facilitará la comunicación (chat, e-mail...) con otras personas (profesores, expertos, compañeros...) o foros?>>



Anexo #9: Planilla Guión de Contenidos

Guión de Contenidos

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos

1. Introducción

1.1 Propósito

<<Especifica el propósito del Guión de Contenidos >>

1.2 Alcance

<<Breve descripción del alcance del Guión; proyectos con los que se relaciona y cualquier otro detalle que pueda facilitar su comprensión. >>

1.3 Referencias

<<Lista completa de todos los documentos referenciados en el Guión. >>

1.4 Resumen

<<Describe qué contiene el Guión. >>



2. Contenidos

<<Debe existir una coherencia con los objetivos, veracidad (diferenciando adecuadamente: datos objetivos, opiniones y elementos fantásticos), profundidad, calidad, organización lógica, buena secuenciación, estructuración (párrafos breves para facilitar su lectura y enlaces con los conceptos relacionados), fragmentación adecuada si se organiza hipertextualmente (para no dificultar el acceso y la comprensión), claridad, actualización, corrección gramatical, ausencia de discriminaciones y mensajes tendenciosos...>>

2.1. Listado de contenidos educativos

<<En esta sección se mostrará un listado de todo el contenido a tratar en el software educativo, organizándolo por materias, con el objetivo de lograr un mejor control y seguimiento de la circulación de los contenidos educativos que interactúan en el modelo de aprendizaje. Los contenidos (conceptuales, procedimentales y actitudinales) que han de trabajar los alumnos se analizarán para descomponerlos en unidades mínimas de presentación, organizarlos y jerarquizarlos en función de su lógica interna, de los niveles de los destinatarios y de los objetivos que deben alcanzar. Considerarán los formatos textuales y audiovisuales. Selección, estructuración y producción de la información (textual, imágenes, sonidos, vídeos, enlaces a otras páginas...). ¿Qué contenidos se presentarán? Selección de conocimientos, nivel de profundidad...>>

2.1.1. Organización de los contenidos

<< ¿Cómo se organizará la información? Secuenciación y estructuración de los contenidos: presentación, índice, apartados y subapartados, resúmenes, hipervínculos... La organización de la materia determinará la estructura modular y la secuenciación de las actividades, y deberá facilitar a los alumnos un aprendizaje significativo y permitir diferentes formas de adquisición de la información; para ello el contenido estará organizado de la siguiente manera:

- De los aspectos más fáciles y concretos a los más complejos y abstractos.
- De los elementos conocidos por los alumnos a los que les son desconocidos.
- De las presentaciones globales o sintéticas a las visiones analíticas.



- *De las visiones episódicas a las sistemáticas.*
- *De los que requieren el uso de habilidades globales a los que implican el uso de habilidades específicas.*
- *Contemplando niveles de dificultad, para facilitar que el alumno escoja el nivel que le interesa y posibilitar que el programa se adapte al nivel de los usuarios. >>*

2.1.2. Redacción de los contenidos

<<Definiciones, explicaciones, multimedias y enlaces que se incluirán en cada caso, hipervínculos... Estilo (coloquial, científico...). Hay que buscar la mejor manera de presentar los conceptos, las imágenes más adecuadas... Todo lo que se incluya debe tener una intencionalidad comunicativa y pedagógica. >>

3. Estrategias de enseñanza/aprendizaje

<<Se diseñarán según una determinada estrategia educativa y teniendo en cuenta los objetivos, los contenidos, los destinatarios y las operaciones mentales que tienen que desarrollar los alumnos. Para ello se seguirán los aspectos que mostraremos a continuación: >>

– **Naturaleza de las actividades educativas**

<<Exposición de información, preguntas, resolución de problemas, búsqueda de información, descubrimiento guiado, descubrimiento experimental. >>

– **Análisis de las respuestas de los alumnos**

<<Es una de las labores más difíciles y meticulosas de los diseñadores, ya que deben prever el mayor número posible de respuestas y, además, tener prevista una "salida" para respuestas imprevistas. >>

– **Estructura**

<<Escenario, elementos relacionados con el contenido, interrelaciones entre ellos. >>



- **Acciones y de respuestas permitidas al alumno.**

- **Duración**

<< Conviene que sea ajustable y no exceda de la capacidad de atención de sus destinatarios. Una sucesión de etapas cortas, con objetivos y contenidos bien definidos, hace que la labor sea más agradable. >>

- **Tipo de control de la situación de aprendizaje que tendrá el alumno**

<< Las actividades que facilitan diversos accesos al mismo material estimulan al alumno a pensar con flexibilidad. >>



Anexo #10: Planilla Diseño de Pruebas Pedagógicas

Diseño de Pruebas Pedagógicas

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1 Propósito

<<Especifica el propósito del Diseño de Pruebas>>

1.2 Alcance

<<Breve descripción del alcance del Diseño; proyectos con los que se relaciona y cualquier otro detalle que pueda facilitar su comprensión. >>

1.3 Referencias

<<Lista completa de todos los documentos referenciados en el Diseño. >>

1.4 Resumen

<<Describe que contiene el Diseño de Pruebas Pedagógicas. >>



2. El programa como objeto material

2.1 Equipo requerido

<<Descripción de los requerimientos de equipos mínimos que exige el programa para funcionar, esta información suele aparecer en los folletos que acompañan al CD, medio de almacenamiento ya usual. >>

2.2 Usabilidad

<<Medida en que el sistema es fácil de aprender y fácil de utilizar. Para ello se analizan los siguientes aspectos:

- **Facilidad de aprendizaje:** Medida en que el usuario novel comprende cómo utilizar inicialmente el sistema y cómo a partir de esta utilización llegar a un máximo nivel de conocimiento y uso del sistema.
- **Flexibilidad:** Multiplicidad de todas formas en las que el usuario y el sistema intercambian información.
- **Solidez:** Características de la interacción que permiten lograr los objetivos, y su asesoramiento.
- **Mecanismos de soporte:** Recursos de ayuda y forma en que el usuario puede ayudarlos. >>

3 El programa como objeto pedagógico

3.1 Contenido

<<El contenido es preciso, actual y adecuado a los alumnos a los que está dirigido el software. El contenido no tiene prejuicios ni estereotipos. >>

- **Contenido científico:** Se trata de evaluar la calidad y la cantidad de la información ofrecida. Los indicadores son los siguientes:



- ✓ **Exactitud, actualidad:** Fechas de edición; referencias o fuentes citadas; términos técnicos; datos estadísticos.
- ✓ **Adecuación:** Valor absoluto (significatividad de los contenidos en sí mismos), valor relativo (adecuación en nivel de tratamiento a la situación pedagógica dada).
- **Contenido socio-cultural e ideológico:** Qué representación de la sociedad encierra el programa; cómo representa otras sociedades. Los indicadores son los siguientes:
 - ✓ **Visión sociocultural:** A qué grupos sociales se refieren los ejemplos, los personajes, los problemas planteados. Qué muestran las ilustraciones: representación racial, género, referencias geográficas, etc.
 - ✓ **Personajes:** Reales, imaginarios; sexo; edad; raza; nacionalidad; condición o estado...
 - ✓ **Situaciones y temas:** Vida cotidiana (en la casa, en la escuela, en el trabajo), situaciones excepcionales (crisis, héroes).
- **Contenido pedagógico:** Se trata de determinar la adecuación pedagógica de los objetivos y contenidos, frente a los usuarios, su nivel y el programa que están desarrollando. Los indicadores son los siguientes:
 - ✓ **Intenciones formativas:** Lo que pretende el programa, los objetivos de aprendizaje que persigue, explícita o implícitamente.
 - ✓ **Niveles de aprendizaje:** Qué niveles de aprendizaje (hechos, conceptos, principios, habilidades, valores) pretende desarrollar el programa.
 - ✓ **Organizadores y autoevaluación:** Contiene síntesis (resúmenes), ejercicios (con o sin respuestas), complementos informativos.



3.2 Comunicación

<<Se trata de evaluar la forma del mensaje, es decir el conjunto de recursos que permiten transmitir un mensaje de un emisor a un receptor. >>

- **Sentido de la comunicación:** Dirección y control de la interacción programa-usuario unidireccional, bidireccional, control de usuario sobre la secuencia, multitareas, multivías...
- **Formas del mensaje:** Los aspectos formales de los códigos elegidos (texto, audio, fotos, animación, gráficos, colores) se justifican en sí y frente a la función que se espera de ellos.

3.3 Método

<<Qué metodología, implícita o explícita, contiene el software para la exposición de las ideas, la organización del trabajo, las formas de uso que determina. >>

- **Organización:** Estructura del manual, forma de exposición y organización de las secuencias.
 - ✓ **Secuencias:** Se componen de una serie de partes que están presentes regularmente.
 - ✓ **Estructura:** El programa es un elemento de enseñanza-aprendizaje.
 - ✓ **Elementos de organización interna:** El programa incluye instrucciones de empleo, índices, objetivos, léxico, preguntas/ejercicios/, respuestas razonadas, recapitulaciones y evaluaciones.
 - ✓ **Papel del maestro:** Se limita a dar instrucciones de uso; es necesario para complementar, aclarar o integrar la información; es hacer un seguimiento del uso y de los logros del estudiante.



- ✓ **Exigencias de aprendizaje:** El programa exige principalmente al estudiante acciones y habilidades para: memorizar información, construir conceptos, seguir instrucciones, construir secuencias de aprendizaje propias, hacer preguntas, construir respuestas originales, relacionar lo aprendido con otros conocimientos.
- ✓ **Distribución de tiempos:** Un estudiante típico, en una sesión de trabajo normal con el programa, distribuye su tiempo aproximadamente en: aprender a navegar y buscar la información desplazándose por el programa, leer texto, preguntas, realizar tareas o ejercicios.
 - **Adaptabilidad:** En qué medida el software impone obligaciones para su uso

4 Calidad educativa

4.1 Creatividad

<<El software estimula la creatividad y permite la toma de decisiones. >>

4.2 Retroalimentación

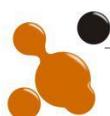
<<La retroalimentación es positiva, oportuna, informativa y pertinente a las respuestas del usuario. >>

4.3 Posibilidades de intervención del docente

<<El docente puede cambiar o añadir contenidos con facilidad. El profesor puede ajustar los parámetros para cada estudiante fácilmente. >>

4.4 Técnicas de preguntas

<<Las preguntas son apropiadas al contenido y miden el dominio del estudiante. >>



5 Calidad técnica

5.1 Claridad

<<Las explicaciones sobre procedimientos e instrucciones son claras. La interfaz es sencilla y de fácil uso. La secuencia de los elementos del menú es lógica. >>

5.2 Gráficos y audio

<<Los gráficos favorecen el centrado de la atención en el contenido y no distraen. Los gráficos y el audio se utilizan para motivar y resultan adecuados a la población, principalmente a los infantes. >>



Anexo #11: Planilla Especificación de Casos de Uso

Especificación de Casos de Uso

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Nombre del Caso de Uso

1.1. Breve descripción

<<La descripción debe ser breve, dejando bien claro el propósito del caso de uso. Un solo párrafo debe ser suficiente para esta descripción. >>

2. Flujo de eventos

2.1. Flujo Básico

<<El caso de uso comienza cuando el actor hace algo. Un actor inicia siempre los casos de uso. El caso de uso debe describir lo que hace el actor y lo que hace el sistema en respuesta. Debe ser expresado bajo la forma de diálogo entre el actor y el sistema. El caso de uso debe describir qué sucede dentro del sistema, pero no cómo o porqué.

Las alternativas simples se pueden presentar dentro del texto del caso de uso. Si toma solamente algunas oraciones para describir qué sucede cuando hay una alternativa, hágala directamente dentro del flujo de sección de los acontecimientos. Si los flujos alternativos son más complejos, utilice una sección separada para describirla. >>



3. Flujo Alterno

<<Alternativas más complejas se describen en una sección separada, mencionada en la subdivisión básica del flujo de sección de los acontecimientos. El uso de flujos alternativos mejora la legibilidad del caso de uso. >>

3.1. Área de funcionalidad

<<A menudo hay flujos alternativos múltiples relacionados con una sola área de la funcionalidad (por ejemplo instalaciones de retiro del especialista, dirección de la tarjeta o recibo dirigido para el caso del uso del efectivo del retirar de una máquina de caja automatizada). Mejora la legibilidad si estos sistemas conceptuales relacionados de flujos se agrupan en su propia subdivisión claramente nombrada. >>

4. Sub-Flujos

<<Un sub-flujo debe ser un segmento del comportamiento dentro del caso de uso que tiene un propósito claro, y es "atómico" en el sentido que usted no hace todas las o ningunas acciones descritas. Usted puede necesitar tener varios niveles de sub-flujos, pero si usted puede, debe evitar esto mientras que se haga el texto más complejo y más duro entender. >>

5. Pre-condiciones

<<Es el estado del sistema que debe estar presente antes de un caso de uso que es realizado. >>

5.1. <Una pre-condición >

6. Post-condiciones

<<Es una lista de estados posibles que el sistema puede tener adentro inmediatamente después de que un caso de uso ha acabado. >>

6.1. <Una post-condición>

7. Requerimientos Especiales

<<Un requisito especial es típicamente un requisito no funcional que es específico a un caso de uso, pero no se especifica fácilmente o naturalmente en el texto del flujo del acontecimiento del caso del uso. Los



ejemplos de requisitos especiales incluyen requisitos legales y reguladores, estándares del uso, y las cualidades de la calidad del sistema de ser construido incluyendo requisitos de la utilidad, de la confiabilidad, del funcionamiento o de la posibilidad de mantenimiento. >>

7.1. <Primer requerimiento especial>

8. Información Adicional

<<Incluye, o proporciona las referencias a cualquier información adicional requerida para aclarar el caso de uso. Esto podría incluir diagramas de descripción, ejemplos o cualquier cosa que usted suponga. >>



Anexo #12: Planilla Plan de Integración

Plan de Integración
<Nombre del Proyecto>
 Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.2 Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3 Referencias

<<Lista completa de todos los documentos referenciados con el Plan de Integración. >>

1.4 Definición, Acrónimos y Abreviaturas

<<En este apartado se debe mostrar las definiciones de todos los términos, siglas y abreviaciones requeridos para entender este documento. >>

1.5 Resumen

<<Describe qué contiene el Plan de Integración. >>



2. Subsistemas

<<En este apartado se debe determinar qué subsistemas se implementarán en esta iteración y establecer el orden en que los subsistemas van a hacer implementados para obtener el tiempo de integración. >>

3. Estructura

<<En este apartado se debe detallar los componentes que se construirán y de cuáles subsistemas formarán parte. De forma individual a cada componente se le debe especificar los criterios de construcción, prueba y cómo será evaluado. >>

3.1. Construcción

<<Describa como se va a llevar a cabo la construcción y cuales componentes serán utilizados en la integración. >>

3.2. Prueba

<<Aquí debe hacer referencia al Plan de Pruebas y señalar cuáles son los casos de prueba, procedimientos de prueba, etc. que van a ser utilizados. >>

3.3. Evaluación

<<En el criterio de evaluación hay que describir las capacidades contra las cuales será evaluado el componente para determinar si ha superado satisfactoriamente el Plan de Pruebas. Dicho plan, en el Criterio de Evaluación, contempla tales aspectos, para ello se hace referencia a todos los aspectos de evaluación que intervienen en esta iteración. >>



Anexo #13: Planilla Plan de Iteración

Plan de Iteración
<Nombre del Proyecto>
 Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito del plan de la iteración. Indique los objetivos en términos mensurables y relaciónese con los riesgos. >>

1.2. Alcance

<<Una breve descripción de lo que se aplica el plan de la iteración; cuál es afectado o influenciado por este documento. >>

1.3. Referencias

<<Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Plan de Iteración. >>



2. Plan

<<Diagramas detallados que demuestran las cronologías, los jalones intermedios, al probar el comienzo, la versión beta, las versiones parciales de programa etc. para la iteración. >>

3. Recursos

<<Los recursos necesitaron para la iteración - humana, financiera, el etc. >>

4. Casos de Uso

<<Listar los casos y los panoramas del uso que se están desarrollando para esta iteración. >>

5. Criterios de evaluación

<<La funcionalidad, funcionamiento, capacidad, medidas de calidad, las metas de calidad, etc. >>



Anexo #14: Planilla Plan de Pruebas

Plan de Pruebas
<Nombre del Proyecto>
 Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar qué se ve afectado por este documento. >>

1.2. Definiciones, Acrónimos y Abreviaturas

<<Especifique las definiciones, abreviaciones y siglas que tienen que ver con este documento a fin de su correcto entendimiento. >>

1.3. Resumen

<<Describe qué contiene el Plan de Pruebas. >>

2. Objetos a ser Evaluados

<<En esta sección se mencionarán todos los componentes y aspectos del sistema que deberán ser evaluados en esta entrega. Existen muchos objetivos de calidad y enfoques para lograrlos. >>



3. Lista de Ideas de las Pruebas

<<En esta sección se exponen las ideas que potencialmente serán las pruebas más útiles a realizar. La Lista de Ideas de las Pruebas ayuda a pensar sobre las pruebas desde etapas muy tempranas y sobre las primeras pruebas a ejecutarse. Es particularmente útil cuando los artefactos están inasequibles o incompletos.

Las pruebas serán identificadas siguiendo la técnica de generación de casos de prueba a través de los casos de uso, detallando los siguientes pasos:

- *Para cada caso de uso, se identifican los caminos posibles, permitiendo establecer los escenarios.*
- *Para cada uno de los caminos, se identifican los conjuntos de valores de entrada y precondiciones, al igual que el resultado esperado.*
- *Se hace, a través de una tabla, un resumen por cada caso de uso que muestre los distintos caminos posibles con sus entradas y salidas.*

Los recursos utilizados para la identificación de las pruebas se mencionan a continuación:

- *El documento de especificación de requerimientos del software.*
- *El documento de arquitectura de software.*
- *Generación de pruebas de sistema a partir de la especificación funcional.*
- *Mejora de la calidad de los requisitos mediante la generación de pruebas.*
- *Especificación e implementación de casos de prueba. >>*

4. Enfoque de las pruebas

<<Esta sección describe cómo las pruebas serán realizadas. Para ello se debe describir para cada tipo de prueba las técnicas a emplear, los propósitos de éstas, los cursos de acción a seguir, los recursos necesarios y las fases que éstas contienen. Entre algunos de los tipos de pruebas tenemos:



Pruebas de Función

Pruebas de datos e integridad de Base de Datos

Pruebas del ciclo del negocio

Pruebas de Interfaces de usuario

Pruebas de Carga

Pruebas de Desempeño

Pruebas de Fallas y Recuperación

Pruebas de Configuración

Pruebas de Volumen

Pruebas de Seguridad y Control de Acceso

Pruebas de Instalación

Señale cada una de los tipos de prueba a emplear en una tabla como la mostrada seguidamente y añada o elimine detalles según corresponda:

T-00: Pruebas de Interfaces de Usuario

Objetivo:	<i>Objetivo del tipo de prueba señalado.</i>
Descripción:	<i>Descripción breve del tipo de prueba.</i>
Técnicas:	<i>Defina el conjunto de procedimientos necesarios para que la prueba se llevada a cabo de manera correcta.</i>
Fases:	<i>1. Criterios de entrada: inicio de la fase 2. Fase 1 3. Fase 2 4. Criterios de salida: fin de la fase 2</i>
Entorno de prueba:	<i>Describa el entorno de prueba, los servicios básicos y configuraciones de la infraestructura requeridos por la prueba.</i>
Hardware:	<i>Detalle la configuración de hardware de los equipos necesarios.</i>
Software:	<i>Señale todos los componentes de software requeridos.</i>



Objetivo:	<i>Objetivo del tipo de prueba señalado.</i>
Configuración:	<i>Mencione el conjunto de peculiaridades específicas para la prueba.</i>
Criterios de Éxito:	<i>Mencione los aspectos que darán muestra que la prueba funcionará.</i>
Consideraciones Especiales:	<i>La prueba puede requerir características especiales, de ser así señale en este apartado las mismas.</i>

5. Herramientas para las Pruebas

<<Esta sección describe las distintas clases de herramientas utilizadas en la prueba del sistema. >>

5.1. Software

<<Se han utilizado las siguientes herramientas durante la prueba:

Herramienta_1: Se trata de una herramienta de utilidades que puede utilizarse para la instalación, configuración y resolución de problemas de las entidades emisoras.

Herramienta_2: Se usa para restaurar y hacer copias de seguridad de archivos. >>

Nombre	Versión	

5.2. Herramientas de Soporte y Productividad

<<Durante las pruebas se utilizaron las siguientes herramientas de supervisión del sistema:

Herramienta_3: Permite ver los contadores, las alertas y los registros de rendimiento del sistema.

Herramienta_4: Se utiliza para comprobar la coherencia de la base de datos. >>

Nombre	Versión	Tipo de herramienta	Descripción



5.3. Hardware

<<Señale cada uno de los dispositivos físicos que comprenden el sistema de computación a utilizar para la realización del conjunto de pruebas. Lo más recomendable que el sistema simule el ambiente de producción. >>

Recurso	Cantidad	Descripción

5.4. Configuraciones de Pruebas de ambiente

<<Las configuraciones del ambiente de Prueba deben ser provistas y soportadas por este proyecto. >>

Nombre de Configuración	Descripción	Implementación de la Configuración Física

6. Casos de Prueba

<<Esta sección detalla las pruebas planeadas a utilizar para comprobar el sistema y garantizar la consecución de sus objetivos con los que fue diseñado. Para realizar estas pruebas se debe llevar a cabo las configuraciones necesarias en el entorno de prueba, cada caso de prueba puede requerir un conjunto de especificaciones determinadas.

Para cada caso de prueba planteado llene una tabla como la presentada a continuación: >>

ID/Nombre/Sistema/Proyecto:	Nivel de Prueba:
ID Caso de Uso:	Tipo(s) de Pruebas(s):
ID Requerimiento: (Si es Caso de Uso no Funcional)	Ambiente de Prueba: (Ubicación)
ID/Nombre Escenario:	Autor del Caso de Prueba:
ID/Nombre Caso de Prueba:	Nombre del Probador:



Versión del Caso de Prueba:	Fecha de Creación:	Fecha de Ejecución:		
Condición(es) para que se ejecute el Caso de Prueba:				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: ___ Fallo: ___ (marque con x el resultado)				
Fecha de Aprobación del Caso de Prueba: _____				

7. Entregables

7.1. Lista de Entregables de Pruebas

<<Especifique en esta sección, los entregables que serán producto de las pruebas a realizar, los cuales serán distribuidos y utilizados por los involucrados en el desarrollo del sistema para brindarles información relevante de los resultados obtenidos y de los avances logrados. >>

Entregables	Descripción
<i>Nombre</i>	<i>Proporcione una breve reseña de la forma y del contenido del entregable.</i>

7.2. Ficha: Escenarios por Caso de Uso

<ID/Nombre de Caso de Uso>



- No existen errores sin solucionar de Gravedad 1 o Gravedad 2.
- No existen errores sin solucionar de Prioridad 1 o Prioridad 2 de ningún nivel de gravedad.
- Todos los casos de prueba del entorno de laboratorio de prueba se han completado satisfactoriamente. >>

8.2. Clasificación de los errores

<<Establezca los criterios de gravedad y prioridad de los errores utilizados en el laboratorio de pruebas. >>

Calificación	Definición de gravedad	Definición de prioridad
1	El error provoca el bloqueo del sistema o la pérdida de datos.	El error debe corregirse lo antes posible. El error bloquea el progreso en esta área.
2	El error causa problemas graves en la funcionalidad u otros aspectos importantes; el producto se bloquea en casos poco claros.	El error debe corregirse antes del lanzamiento del producto.

8.3. Resultados de la prueba

<<Se obtuvieron resultados positivos de todos los casos de prueba. No hubo errores sin resolver de Gravedad 1 y 2 ni de un nivel de Prioridad 2 o superior. Esto demuestra la consecución de los objetivos de las pruebas, con lo que la solución está lista para el lanzamiento. >>

9. Riesgos

<<Enumere cualquier riesgo que pueda afectar la ejecución de este plan de pruebas, e identifique las estrategias de la mitigación, contingencia y el impacto que pueden generarse por cada riesgo. Para



detalles muy específicos emplee el Plan de Gestión de Riesgos, comente aquí solo los riesgos a manera general. >>

Riesgos	Estrategias de Mitigación	Plan de Contingencia	Impacto
<i>Nombre</i>	<i>Estrategia 1.</i>	<i>Actividad 1</i>	<i>Indique las áreas que se podrían ver afectadas por el riesgo</i>
	<i>Estrategia 2.</i>	<i>Actividad 2</i>	
	<i>Estrategia N.</i>	<i>Actividad N</i>	



Anexo #15: Planilla Manual de Instalación

Manual de Instalación

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito del Manual de Instalación. >>

1.2. Alcance

<< Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<< Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Manual de Instalación. >>



2. Requerimientos Mínimos del Sistema

<<Aquí se debe ver reflejado lo que requiere el sistema para que sea instalado y utilizado correctamente. Esto incluye: el procesador de sistema, la memoria de sistema, espacio libre en disco, sistema operativo, software, acceso a la red, etc. Llenar la tabla que se muestra a continuación y añada o elimine requerimientos según sea necesario: >>

Procesador de Sistema:	<i>Mencione la velocidad de trabajo del procesador de sistema. Esta se mide en Megahertzios (MHz).</i>
Memoria de Sistema:	<i>Indique la capacidad de la memoria requerida para instalar el sistema (Ej. 256 MB).</i>
Espacio Libre en Disco:	<i>Refleje el espacio libre existente en el disco (Ej. 10 MB).</i>
Sistema Operativo:	<i>Indique el sistema operativo utilizado.</i>
Tarjeta Gráfica:	<i>Indique compatibilidad y la capacidad de memoria mínima (Ej., 64 MB).</i>
Software Existente:	<i>Mencione los programas utilizados para el sistema.</i>
Acceso Red:	<i>Coloque el acceso a la red (Ej. Acceso a Internet).</i>

3. Requerimientos Recomendados del Sistema

<<Se debe reflejar los requerimientos que recomienda la organización, para que el sistema sea instalado y utilizado con un mejor rendimiento. Esto incluye: el procesador de sistema, la memoria de sistema, espacio libre en disco, sistema operativo, software, acceso a la red, etc. Llenar la tabla que se muestra a continuación y añada o elimine requerimientos según sea necesario: >>



Procesador de Sistema:	<i>Mencione la velocidad de trabajo del procesador de sistema. Esta se mide en Megahertzios (MHz).</i>
Memoria de Sistema:	<i>Indique la capacidad de la memoria requerida para instalar el sistema (Ej. 512 MB).</i>
Espacio Libre en Disco:	<i>Refleje el espacio libre existente en el disco (Ej. 200 MB).</i>
Sistema Operativo:	<i>Indique el sistema operativo utilizado.</i>
Tarjeta Gráfica:	<i>Indique compatibilidad y la capacidad de memoria mínima (Ej., 128 MB).</i>
Software Existente:	<i>Mencione los programas utilizados para el sistema.</i>
Acceso Red:	<i>Coloque el acceso a la red (Ej. Acceso a Internet).</i>

4. Instalación

<<Aquí se tiene que detallar las instrucciones de instalación. Se debe probar esas instrucciones realizándolas en un equipo de prueba. Se debe responder preguntas como: ¿Qué otro software debe ser instalado primero?, ¿Cómo instalo el producto?, ¿Cómo desinstalo el producto? y ¿Qué hago si encuentro otros problemas? >>

4.1. Instalar otro Software

<<Mencionar que software debe ser instalado antes de instalar el producto, para que este funcione correctamente. >>

4.2. Instalar el Producto

<<Debe reflejarse los pasos y subpasos para instalar el producto. >>



4.3. Desinstalar el Producto

<<Debe reflejarse los pasos y subpasos para desinstalar el producto. >>

4.4. Problemas de Instalación

<<En este apartado se debe reflejar la solución a los problemas que se pueden presentar durante la instalación del producto. Pueden mostrarse preguntas como:

Veo un mensaje de error que dice "mensaje-de-error". ¿Qué está mal?

Respuesta.

No puedo realizar "tarea-común". ¿Qué está mal?

Respuesta. >>

5. Empezar a utilizar el Producto

<<Se debe reflejar los pasos para correr una prueba unitaria y empezar a utilizar el producto. >>

5.1. Prueba Unitaria

<<Para versiones de desarrollo o componentes reutilizables, incluya instrucciones para correr pruebas unitarias. Indicar los pasos para correr una prueba unitaria después de la instalación.

5.2. Utilizar el Producto

<<Indicar qué se deberá hacer para empezar a utilizar rápidamente el producto. >>



Anexo #16: Planilla Manual de Usuario

Manual de Usuario

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito del Manual de Usuario. Indique los objetivos en términos mensurables y relaciónese con los riesgos. >>

1.2. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<< Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Manual de Usuario. >>



2. Contenido

<<Este apartado se debe llenar con la información del manual del usuario que se encuentra abajo, según corresponda con su proyecto. El manual tiene que realizarse paso-por-paso y colocar material de referencia. Puede organizarse por características, módulos, por casos de uso, por roles o de otras formas, que contribuyan al claro entendimiento por parte de los usuarios de las funciones y herramientas con que cuenta el sistema. >>

2.1. <Nombre de la Sección>

2.1.1. <Nombre de la Subsección>



Anexo #17: Planilla Plan de Adiestramiento

Plan de Adiestramiento

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito del Plan de Adiestramiento. Indique los objetivos en términos mensurables y relaciónese con los riesgos. >>

1.2. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<<Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Plan de Adiestramiento. >>



2. Consideraciones Generales

2.1. Objetivos del Plan de Entrenamiento

<<Mencionar los objetivos del Plan de Entrenamiento. >>

2.2. Programas

<<Se debe reflejar a quiénes van dirigidos los programas, los eventos (seminarios, cursos, etc.) contenidos en cada programa y la importancia de los temas a discutir en cada eventos. Por ejemplo:>>

Programa dirigido a <Nombre>

Evento 1.....n

2.3. El Perfil de los Instructores

<<Reflejar las características de los instructores exigidas por la organización. Se puede tomar en cuenta el siguiente texto de ejemplo:

Los instructores de los programas deben ser expertos en los temas de cada uno de los seminarios y cursos a ser dictados. Deberán contar con una amplia experiencia académica y laboral, y haber prestado sus servicios profesionales en el tema, en instituciones del sector público. Se requiere, además, que sean o hayan sido docentes de reconocidas universidades del país, y que hayan realizado publicaciones sobre el tema, o temas afines. >>

2.4. Tipo de Eventos

<<En este apartado se debe responder las preguntas siguientes: ¿Cómo está conformado cada programa? ¿Cuáles son las características de los eventos? ¿Quiénes pueden ser invitados a los eventos? ¿Cuál es la modalidad utilizada para la realización de los eventos? >>

2.5. Incentivos para la participación

<<Reflejar los instrumentos que la organización deberá establecer para incentivar al personal a participar en los diferentes eventos de entrenamiento (seminarios, cursos, etc.). >>



2.6. Difusión y Ejecución

<<En este apartado se le debe dar respuesta a las siguientes preguntas: ¿Qué medios serán utilizados por la organización para informar sobre el Plan de Adiestramiento? ¿Qué unidad organizativa se encargará de la difusión y ejecución del Plan? ¿Qué unidad organizativa se encargará del seguimiento, control, evaluación y reajuste del Plan? >>

2.7. Mecanismos de Selección y Evaluación

<<Mostrar las características que serán tomadas en cuenta para la selección de los participantes, así como también los criterios de evaluación. El siguiente texto puede servir de guía:

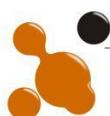
El Plan de Adiestramiento pretende alcanzar la mayor participación de los recursos humanos en los eventos. Sin embargo, de ser necesario se deberían considerar características como:

- *Funciones y actividades que desempeña el participante.*
- *Formación académica.*
- *Experiencia laboral.*
- *Responsabilidad, creatividad, capacidad crítica, autoaprendizaje.*
- *Capacidad para investigar problemas concretos de la provincia y plantear.*
- *Soluciones viables.*

Por otro parte, los seminarios y cursos que contiene el Plan deben ser evaluados considerándolas evaluaciones de desempeño, de impacto, y permanente.

Al interior de cada una de estas evaluaciones es necesario analizar lo siguiente:

- *Evaluación del desempeño: diseño de procedimientos e instrumentos de medición y recolección de datos; medición del desempeño actual; y, análisis evaluativo del desempeño individual y grupal.*



- *Evaluación de impacto: análisis comparativo de la productividad y determinación de la variación de la productividad derivada de la capacitación recibida.*
- *Evaluación permanente: evaluación de seguimiento; evaluación de desempeño; e informes periódicos. >>*

3. < Nombre del Programa>

3.1. < Nombre del Evento>

3.1.1. Justificación

<<Justificar por qué es necesaria la realización de este evento. En este apartado se deben exponer todos los problemas existentes que conllevan a la ejecución del evento. >>

3.1.2. Objetivos

<<Mencionar los objetivos tanto del evento (curso, seminario, etc.) como los objetivos del aprendizaje. >>

3.1.3. Temario

<<Mostrar el contenido del evento. Se debe realizar una lista que muestre de forma organizada y secuencial los aspectos a hacer tratados en el evento. >>

3.1.4. Destinatarios

<<Especificar a quiénes está dirigido el evento (curso, seminario, etc.). >>

3.1.5. Perfil del Instructor

<<Reflejar las características que deberá poseer el instructor del evento según las exigencias de la organización. >>

3.1.6. Tipo de Evento

En este apartado se debe responder las preguntas siguientes: ¿Cuáles son las características del evento? ¿Quiénes pueden participar en el evento? ¿Cómo se realizará el evento? El siguiente texto de ejemplo puede servir de apoyo:



El curso se llevará a cabo de forma virtual. Al inicio del curso se remitirá material docente (en formato electrónico) a los participantes, para su estudio.

El temario del curso se desarrollará a través de sesiones en las que se combinará la presentación del marco conceptual, teórico y metodológico, mediante exposiciones teóricas, con el análisis y discusión de casos prácticos.

En el desarrollo de las sesiones se estimulará la participación activa de los alumnos con el propósito de dotar al curso de una orientación eminentemente práctica.

Es necesaria la asistencia a un mínimo de 80% de las horas programadas, y cada cursante presentará, para la obtención del certificado de participación, una pequeña monografía, entre 10 y 15 páginas, de mejora de su actividad laboral en relación con el objeto del curso.



Anexo #18: Planilla Plan de Implantación

Plan de Implantación
<Nombre del Proyecto>
Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito del Plan de Implantación. Indique los objetivos en términos mensurables y relaciónese con los riesgos. >>

1.2. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<<Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Plan de Implantación. >>



2. Actividades de entrega del producto

<<Se debe describir todas las actividades que son necesarias para la entrega del producto, que incluye la planificación, las pruebas betas, los elementos a hacer entregados, la instalación y el soporte. >>

2.1. Cronograma

<<El cronograma tiene el propósito de definir y ordenar cronológicamente las actividades, en este caso las de la implantación del sistema. En este apartado hay que describir el cronograma y los puntos de control que conllevan a las actividades de la implantación. Los puntos de control de la implantación deben concordar con los del proyecto. Se debe reflejar los siguientes aspectos: la planificación de la implantación, desarrollo del material de soporte, pruebas de aceptación en el lugar de implantación, gerencia de las pruebas de aceptación, gerencia de los proyectos pilotos o programas betas, gerencia de la Masificación de la implantación del producto y empaquetado y hacer el producto accesible desde Internet. >>

2.2. Responsabilidades

<<Identificar las responsabilidades que tiene el cliente y el equipo de desarrollo en la preparación para la implantación, describir el nivel de responsabilidad del usuario/cliente en las pruebas de aprobación del producto y para manejar cualquier anomalía. >>

3. Tipo de Implantación

<<Llene la información que ayudará a otras personas a entender este sistema en un vistazo. Siéntase libre de usar información técnica relevante y nombre plataformas tecnológicas específicas. >>

Tipo de sistema:	<p>Aplicación Gráfica de escritorio</p> <p>Comando tipo Unix</p> <p>Aplicación Web de servidor</p> <p>Servicio Web</p>
------------------	--



	<p><i>Applet del lado del cliente</i></p> <p><i>Aplicación incrustada</i></p> <p><i>Librería reutilizable</i></p> <p><i>Estructura de clases reutilizable</i></p> <p><i>Plugin para navegador</i></p>
Lenguaje(s) de Programación:	<p><i>Java</i></p> <p><i>Perl, scripts de shell para Unix</i></p>
Almacenamiento de Datos:	<p><i>Archivos planos usando XML.</i></p> <p><i>Archivos planos usando el formato de propiedades de archivo de Java.</i></p> <p><i>Archivos planos usando el formato de serialización de objetos de Java.</i></p> <p><i>Base de datos SQL: MySQL.</i></p>
Tecnologías para UI:	<p><i>Java Swing</i></p> <p><i>XHTML, CSS, JavaScript</i></p>
Tecnología para Seguridad:	<p><i>Autenticación: No es necesaria</i></p> <p><i>Autenticación: Archivo local con nombre de usuario y contraseña</i></p> <p><i>Autenticación: LDAP</i></p> <p><i>Autorización: Permisos de archivos del sistema operativo y</i></p>



	<i>banderas de lectura-escritura-ejecución</i> <i>Autorización: Listas de control de acceso</i> <i>Encriptación: No es necesaria</i> <i>Encriptación: SSL</i>
--	--

4. Recursos y Fuentes

<<Se debe describir los recursos y fuentes que son necesarios para realizar las actividades previstas en la implantación. >>

4.1. Hardware

<<En este apartado hay que identificar el hardware necesario para apoyar el software y detallar el modelo, las versiones y las configuraciones. Aquí debe verse reflejada la información del fabricante del hardware y también información de soporte. >>

4.2. Facilidades

<<Describir todos los recursos que son necesarios para probar e instalar el sistema. Incluye: construcciones civiles especiales o cuartos con doble piso, requerimientos de energía particulares, y características especiales para soportar los requerimientos de seguridad y privacidad. >>

4.3. Implantación

<<Aquí se describe la documentación, el personal, el software y las características específicas necesarias para soportar el producto que se va a entregar. >>

4.4. Personal de Apoyo

<<En caso de que aplique, se debe describir el personal y sus niveles de capacitación que son necesarios para apoyar el producto que se va a entregar. >>



4.5. Documentación de Apoyo

<<En caso de que aplique, se debe describir la documentación que es requerida para apoyar al producto a entregar, como: procedimientos y casos de prueba, manuales de usuario, etc. >>

4.6. Software de Apoyo

<<En caso de que aplique, se debe hacer una descripción de todo el software necesario para soportar al producto que va hacer entregado. Esto incluye compiladores, herramientas, datos de prueba, utilidades, archivos de datos, etc. >>



Anexo #19: Planilla Pautas del Diseño Gráfico

Pautas del Diseño Gráfico

<Nombre del Proyecto>

Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción

1.1. Objetivos

<<Una breve descripción del propósito de las Pautas del Diseño Gráfico. Indique los objetivos en términos mensurables y relaciónese con los riesgos. >>

1.2. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<<Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene las Pautas del Diseño Gráfico. >>



2. Diagramación general

<<En esta sección aparecerá una imagen donde se identifiquen las diferentes áreas que componen de manera general a todas las pantallas (Títulos, menús, ventanas, íconos, botones, espacios de texto-imagen, formularios, barras de navegación, barras de estado, elementos hipertextuales, fondo, etc.) >>

3. Dimensiones

<<En esta sección aparecerá una imagen a línea de cada pantalla tipo, superpuesta a la imagen a todo color de la pantalla tipo correspondiente (50% de opacidad) donde se especifiquen las dimensiones de las mismas mediante cotas. >>

4. Pauta Cromática

<<En esta sección se define con una muestra (cuadrado) de cada color utilizado y junto a cada muestra debe aparecer el código hexadecimal o RGB del color. Se debe mostrar alguna pantalla tipo (a todo color) donde aparezcan representados los colores utilizados. >>

5. Pauta Tipográfica

<<En esta sección se define, en una lista con los nombres de todas las tipografías empleadas en el software, utilizando para esto la misma tipografía que se está nombrando. Se debe especificar de cada una de ellas el modo o estilo (cursiva, negrita, etc.), el puntaje (tamaño) y el color en todos los lugares donde se empleen. Se debe mostrar al menos una pantalla tipo donde convivan todas o la mayoría de las tipografías utilizadas. Se deben utilizar tipografías estándares establecidas para software. >>

6. Estilo de Menú y Botones

<<Se define el tipo de menú, desplegable, si va a aparecer por disolvencia, es decir, todas sus características. Se mostrará una o varias imágenes (según sea necesario) que representen todas las características del menú. Se define si los botones van a ser solo texto, si convivirán sobre una imagen o si serán simplemente una imagen. Sea cual sea el tipo de botón deberán pautarse los diferentes estados del mismo (reposo, sobre y presionado), esto se hará mostrando las imágenes correspondientes de cada estado para cada uno de los tipos de botones utilizados. >>



7. Estilo de Imágenes

<<Se define el estilo de las imágenes que se utilizarán (blanco y negro o a todo color) y además el formato (.png, .jpg, etc.) y las dimensiones de las mismas. >>

8. Pautas de Formularios y Tablas

<<Se define el estilo que tendrá cada tabla empleada en el sitio, tablas donde las divisiones de las celdas serán líneas, por diferencia de planos de color o empleando cualquier otro recurso gráfico. >>

9. Pautas para el Diseño de Banners

<<Se define el formato de los banners (.gif, .swf, etc.), las dimensiones, la composición (imágenes, texto, sonido), el estilo de las imágenes a emplear (a color, blanco y negro, vectoriales, etc.). >>

10. Pautas de Impresión

<<Se define las áreas o contenidos imprimibles del software y su formato de impresión. >>

11. Prototipos de Pantallas

<<Se deben mostrar tantas pantallas como sean necesarias para tener completa claridad del diseño del software en su totalidad. >>

12. Recortes

<<En este documento hemos llamado recortes a los elementos gráficos que se generan, es decir a la materia prima que se produce como resultado del trabajo de diseño gráfico y que se necesita y es indispensable para el montaje del software.

Banners.

Iconos.

Cabezales y Pie de Página.

Imágenes de fondo. >>



13. Elementos gráficos

<<Los elementos gráficos no forman parte del documento, pero se deben entregar organizados en una carpeta y nombrados de forma coherente de manera tal que exista una correspondencia entre los nombres y la función del objeto en el software. >>



Anexo #20: Planilla Plan Audiovisual

Plan Audiovisual
<Nombre del Proyecto>
Versión <x.x>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Tabla de Contenidos:

1. Introducción**1.1. Objetivos**

<<Una breve descripción del propósito del Plan Audiovisual. >>

1.2. Alcance

<<Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento. >>

1.3. Referencias

<<Una lista de documentos relacionados o referidos. >>

1.4. Resumen

<<Describe qué contiene el Plan Audiovisual. >>



2. Plan para las medias

<<En esta sección los recursos educativos de la aplicación serán referenciados empleando un código identificador único en todo el guión. Este nombre se conforma de la siguiente manera:

La letra que identifica el tipo de recurso + número consecutivo para diferenciar a uno del otro.

Ej. I34 – se refiere a la imagen 34, V22- Se refiere al video 22, PC13, se refiere a la palabra caliente No 13, etc.

A continuación la tabla de relación entre las letras y el tipo de recurso. >>

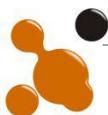
Tipo de recurso	Identificador
Imagen fija	I
Sonidos	S
Videos	V
Animaciones	A
Locuciones.	L
Palabras Calientes	PC

3. Plan de Adquisición

<<Esta sección tiene como objetivo la caracterización necesaria de la localización física de cada una de las medias. Todas las planillas tendrán los siguientes campos generales:

Referencia	Descripción	Localización

- La casilla de referencia se refiere a un identificador único conformado de la siguiente manera: Letra que identifica el tipo de recurso + un número consecutivo. (Ej. I04 – se refiere a Imagen 4, V05- se refiere a video 5, etc.)



- En la casilla de descripción se describe la media para su identificación.
- En ocasiones se define una columna localización, cuando se tienen los recursos que se desean, sin embargo, estos están sujetos a revisión de un especialista para comprobar que cumple con los requerimientos de calidad según su utilidad para el software.

4. Especificación por tipos de media

Imagen fija (I)

Ref.	Descripción

Sonidos (S)

Ref.	Descripción	Tipo de sonido	Categoría

- En la casilla Tipo de sonido deberá indicarse si es: Efecto sonoro, música.
Si se marca la casilla Música entonces se llenarán las casillas: Título y Categoría.
- La casilla Categoría se refiere a si es la canción infantil o no, o instrumental o cantada.

Videos. (V)

Ref.	Descripción



Animaciones. (A)

<i>Ref.</i>	<i>Descripción</i>

Nota: Se debe especificar si hay locuciones asociadas e identificarlas.

Locuciones (L)

<i>Ref.</i>	<i>Texto.</i>	<i>Personajes que intervienen</i>

En esta media se escriben los textos que deberán ser grabados para apoyar las animaciones, diaporamas, videos, etc. (entiéndase también como locuciones voz en off)

- En la casilla Texto se escribe el texto que se dice en la locución.*
- En la casilla Personajes que intervienen se debe especificar bien si es la voz de una o más personas y qué tipo de voz (si la de un niño, un adulto o impersonificada).*



Anexo #21: Modelo para informar el peso de los criterios

1. Para esta investigación a usted se le pide otorgar el nivel de importancia que considera, tiene cada criterio dentro del grupo y debe asignarle una puntuación en dependencia del valor total dado al grupo al que pertenece:

Grupo No. 1: Criterios de Valor científico (Total a asignar 25)

- ✓ Valor científico de la investigación. Puntos ____
- ✓ Calidad de la investigación desarrollada. Puntos ____
- ✓ Novedad científica de la investigación. Puntos ____

Grupo No. 2: Criterios de implantación (Total a asignar 25)

- ✓ Necesidad real de implantación de la propuesta. Puntos ____
- ✓ Posibilidad de ser aplicada la propuesta. Puntos ____
- ✓ Integración de la propuesta con el proceso de desarrollo de SWE en la UCI. Puntos ____

Grupo No.3: Criterios de adaptación (Total a asignar 20)

- ✓ Adaptabilidad de la propuesta al proceso de desarrollo de SWE en la UCI. Puntos ____
- ✓ Integración de la propuesta con la metodología de desarrollo utilizada. Puntos ____
- ✓ Integración con el entorno de producción donde se aplique. Puntos ____

Grupo No.4: Criterios de impacto (Total a asignar 30)

- ✓ Repercusión de la propuesta en el proceso de desarrollo de SWE en la UCI. Puntos ____
- ✓ Organización en el proceso de documentación de la investigación. Puntos ____
- ✓ Fluidez de los procesos de desarrollo de SWE en la UCI con el uso de la propuesta. Puntos ____



Anexo #22: Modelo para calificar los criterios.

1. A usted se le pide una evaluación de los criterios para la Propuesta de documentación para el proceso de desarrollo del software educativo presentada en esta investigación utilizando una escala del 1 al 5 (1 significa Muy Mal ,2 Mal, 3 Regular ,4 Bien y 5 Excelente).

Grupo No. 1: Criterios de Valor científico.

- ✓ Valor científico de la investigación. Evaluación____
- ✓ Calidad de la investigación desarrollada. Evaluación____
- ✓ Novedad científica de la investigación. Evaluación____

Grupo No. 2: Criterios de implantación.

- ✓ Necesidad real de implantación de la propuesta. Evaluación____
- ✓ Posibilidad de ser aplicada la propuesta. Evaluación____
- ✓ Integración de la propuesta con el proceso de desarrollo de SEE en la UCI. Evaluación____

Grupo No.3: Criterios de adaptación.

- ✓ Adaptabilidad de la propuesta al proceso de desarrollo de SWE en la UCI. Evaluación____
- ✓ Integración de la propuesta con la metodología de desarrollo utilizada. Evaluación____
- ✓ Integración con el entorno de producción donde se aplique. Evaluación____

Grupo No.4: Criterios de impacto.

- ✓ Repercusión de la propuesta en el proceso de desarrollo de SWE en la UCI. Evaluación____
- ✓ Organización en el proceso de documentación de la investigación. Evaluación____
- ✓ Fluidez de los procesos de desarrollo de SWE en la UCI con el uso de la propuesta.
Evaluación____

2. Categoría final que le asigna a la propuesta:

Excelente (Alta novedad científica, con aplicabilidad y resultados relevantes) ____

Bueno (Novedad científica, resultados destacados) ____

Aceptable (Suficientemente bueno con reservas) ____

Cuestionable (No tiene relevancia científica y los resultados son malos) ____



Malo (No aplicable) ____

3. Valoración final de la propuesta.

Sugerencias del evaluador para mejorarla:

Elementos críticos que deben mejorarse:

